

PROYECTO DE FINAL DE CARRERA:

**Estudio de la sensibilidad de
detectores QRS frente a cambios de
morfología en el ECG**

Autor: Alejandro Chuang Yang

Tutor: Miguel Ángel García González

ETSETB - UPC

*... gracias a mi familia por la confianza
que en mi siempre ha depositado...
y un entrañable recuerdo a todos mis amigos...*

0. **Índice**

1. Introducción.....	pág. 7
2. Objetivos del proyecto.....	pág. 20
3. Descripción de los algoritmos implementados.....	pág. 21
3.1 Detector de ECGs mediante la transformada Wavelet.....	pág. 22
3.2 Detector de ECGs mediante la algoritmos genéticos.....	pág. 33
3.3 Detector de ECGs mediante la transformada de Hilbert.....	pág. 39
3.4 Detector de ECGs mediante cruces por cero.....	pág. 43
4. Evaluación y comparación de los diferentes detectores.....	pág. 52
5. Creación de ECGs sintéticos.....	pág. 57
6. Rotación de los electrocardiogramas sintéticos.....	pág. 63
7. Detección de complejos QRS a partir de ECGs sintéticos rotados.....	pág. 72
8. Eliminación de artefactos tras la detección.....	pág. 75
9. Limitación de la casuística.....	pág. 80
10. Estudio sobre todos los pacientes de la base de datos.....	pág. 82
11. Corrección de las detecciones mediante el detector por correlación.....	pág. 86
12. Aplicación del proceso de detección de complejos QRS a pacientes transplantados de corazón.....	pág. 109
13. Conclusiones.....	pág. 118
14. Bibliografía.....	pág. 122
15. Anexo.....	pág. 123

1. Introducción

EL ELECTROCARDIOGRAMA (ECG) es una señal que se usa comúnmente en medicina, principalmente, por su simple proceso de adquisición y por la información que se puede diagnosticar a partir de él sobre las diferentes patologías del corazón. Representa la actividad eléctrica del corazón sobre la superficie del cuerpo humano.

Fue en 1889 cuando el francés Augustus Desiré Waller logró, por primera vez, obtener un ECG por medio de un electrómetro capilar. Posteriormente, en 1903, Einthoven midió algunas anormalidades cardíacas gracias a un galvanómetro de hilo. Asimismo, introdujo conceptos en electrocardiografía como el etiquetado de varias ondas ECG usando las letras de la P a la U, la colocación de electrodos para algunos tipos de medidas así como la definición de un primer modelo cardíaco teórico. Además, Einthoven fue el primero en emplear el acrónimo “EKG” (en vez de “ECG” puesto que la palabra “cardio” en holandés se escribe con “k”).

Las primeras aplicaciones por ordenador en tecnología médica fueron las de procesado y análisis de la señal ECG. Los primeros sistemas ECG computerizados consistían en un marco que procesaba la señal y detectaba sus principales características.

Debido al hecho de que la actividad del corazón es rítmica y totalmente ordenada, la forma de onda que se obtiene de ésta también será regular (ver figura 1). En ella se reconocen fácilmente diversas ondas cuyas amplitudes, duraciones y morfologías están bien definidas. Las ondas de mayor relevancia son:

-onda P: es debida a la despolarización de la musculatura de las aurículas y se caracteriza por tener una duración menor a 0.12 s y una amplitud menor a 0.25 mV. En esta onda se pueden ver la presencia de arritmias.

-ondas Q, R y S: forman el complejo QRS y se deben a la despolarización de los ventrículos y a la repolarización de las aurículas. Se trata de la componente de mayor amplitud del ECG (variando entre los 0.6 y los 1 mV) y su duración es menor a 0.12 s. En él se pueden apreciar infartos, trastornos de la conducción sanguínea, etc.

-onda T: tras el complejo QRS, tiene lugar dicha onda de repolarización ventricular, que expresa la vuelta de los ventrículos al estado eléctrico de reposo inicial. La duración de la onda T no supera los 0.2 s y su amplitud varía entre los 0.2 y los 0.4 mV.

-onda U: a menudo no es observable; se debe a potenciales residuales del músculo ventricular y a la repolarización lenta de los músculos papilares.

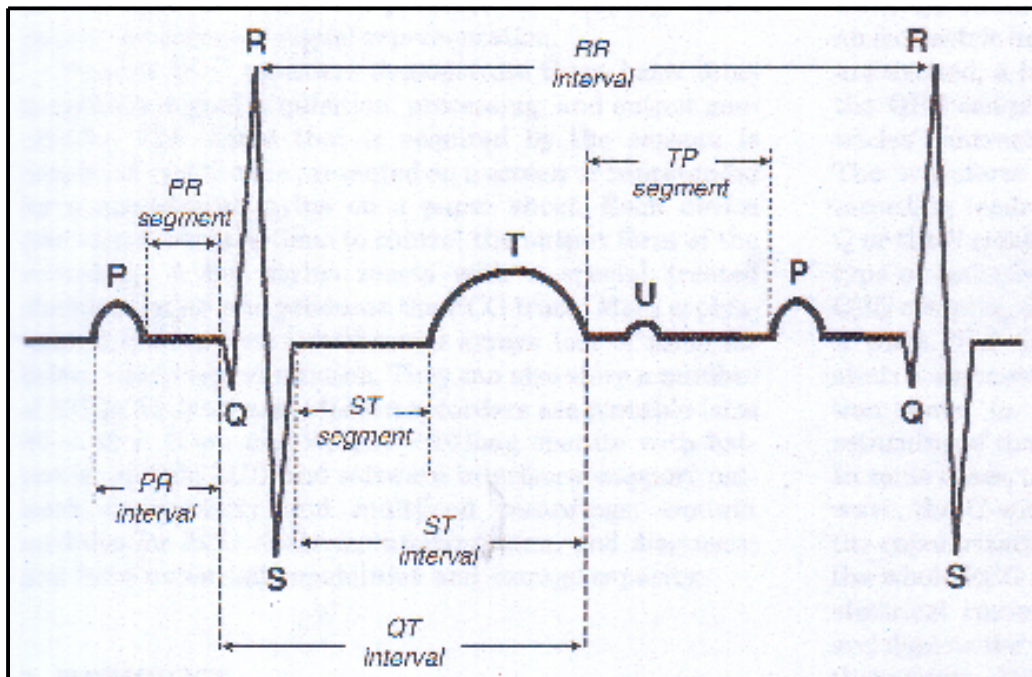


Fig. 1 Características del ECG empleado en exámenes clínicos¹.

LA ADQUISICIÓN de la forma de onda ECG puede ser llevada a cabo mediante una medida diferencial de amplitudes entre dos puntos o derivaciones. Einthoven usó tres derivaciones llamadas I, II y III (figura 2) que definió de la siguiente manera: $I = V_{LA} - V_{RA}$, $II = V_{LL} - V_{RA}$, $III = V_{LL} - V_{LA}$ (donde LA equivale a “left arm” (brazo izquierdo), RA a “right arm” (brazo derecho) y LL a “left leg” (pierna izquierda)). Puesto que el cuerpo humano es un muy buen conductor de electricidad, los electrodos en las derivaciones superiores pueden colocarse en el torso, cerca de las espaldas y de las caderas sin que haya ninguna pérdida de información de la actividad eléctrica del corazón.

En 1934, Wilson introdujo el concepto de medidas “unipolares” combinando las tres extremidades y definiendo su potencial medio como punto de referencia. De este modo, cada derivación puede medir la amplitud diferencial con respecto a la misma referencia. A partir de entonces, la medida estándar del ECG consistiría en la medida de 12 derivaciones que definirán los estándares de la electrocardiografía (figura 2): las tres derivaciones de las extremidades de Einthoven (I, II, III), las tres derivaciones

aumentadas (aVL, aVR, AVF) y las seis derivaciones unipolares (o torácicas) (V_{1...6}). Las derivaciones aumentadas que Wilson introdujo fueron:

$$aVL = \frac{2V_{LA} - V_{RA} - V_{LL}}{2}$$

$$aVR = \frac{2V_{RA} - V_{LA} - V_{LL}}{2}$$

$$aVF = \frac{2V_{LL} - V_{LA} - V_{RA}}{2}$$

Las derivaciones unipolares se definen con la siguiente fórmula:

$$V_i = u_i - \frac{1}{3}(V_{LA} + V_{RA} + V_{LL})$$

Donde $i = 1, 2, \dots, 6$ y u_i es el potencial medido en cada punto del pecho.

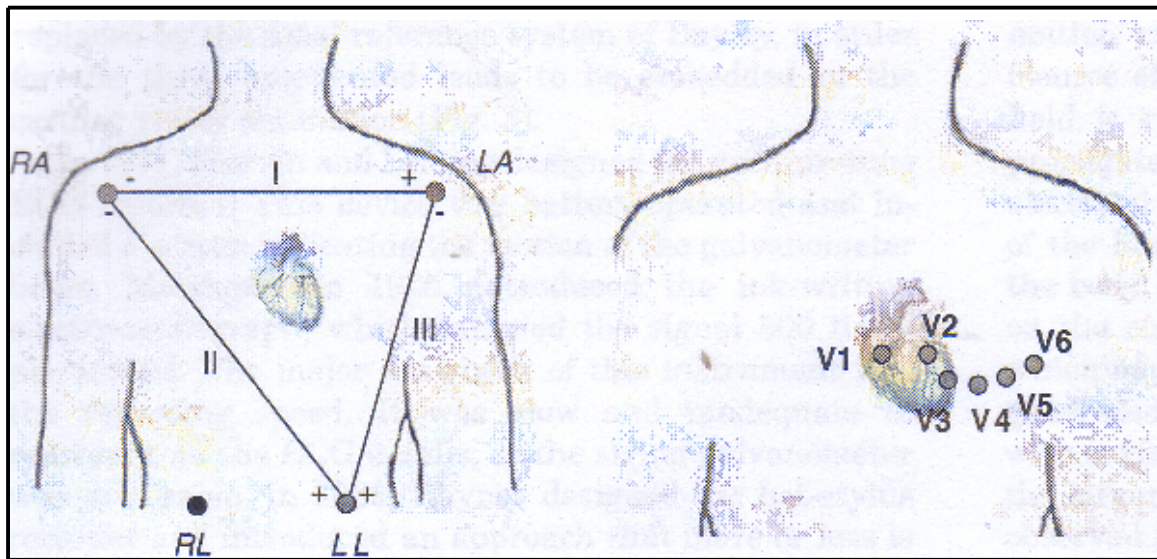


Fig. 2 Las 12 derivaciones del ECG¹.

Tal calado tuvo dicho concepto de las medidas unipolares que hasta el día de hoy, como podemos observar en la siguiente figura, se siguen aplicando sus directrices. En el siguiente electrocardiograma podemos ver las derivaciones I, II y III así como las medidas aVR, aVL, aVF, V1, V2, V3, V4, V5 y V6.

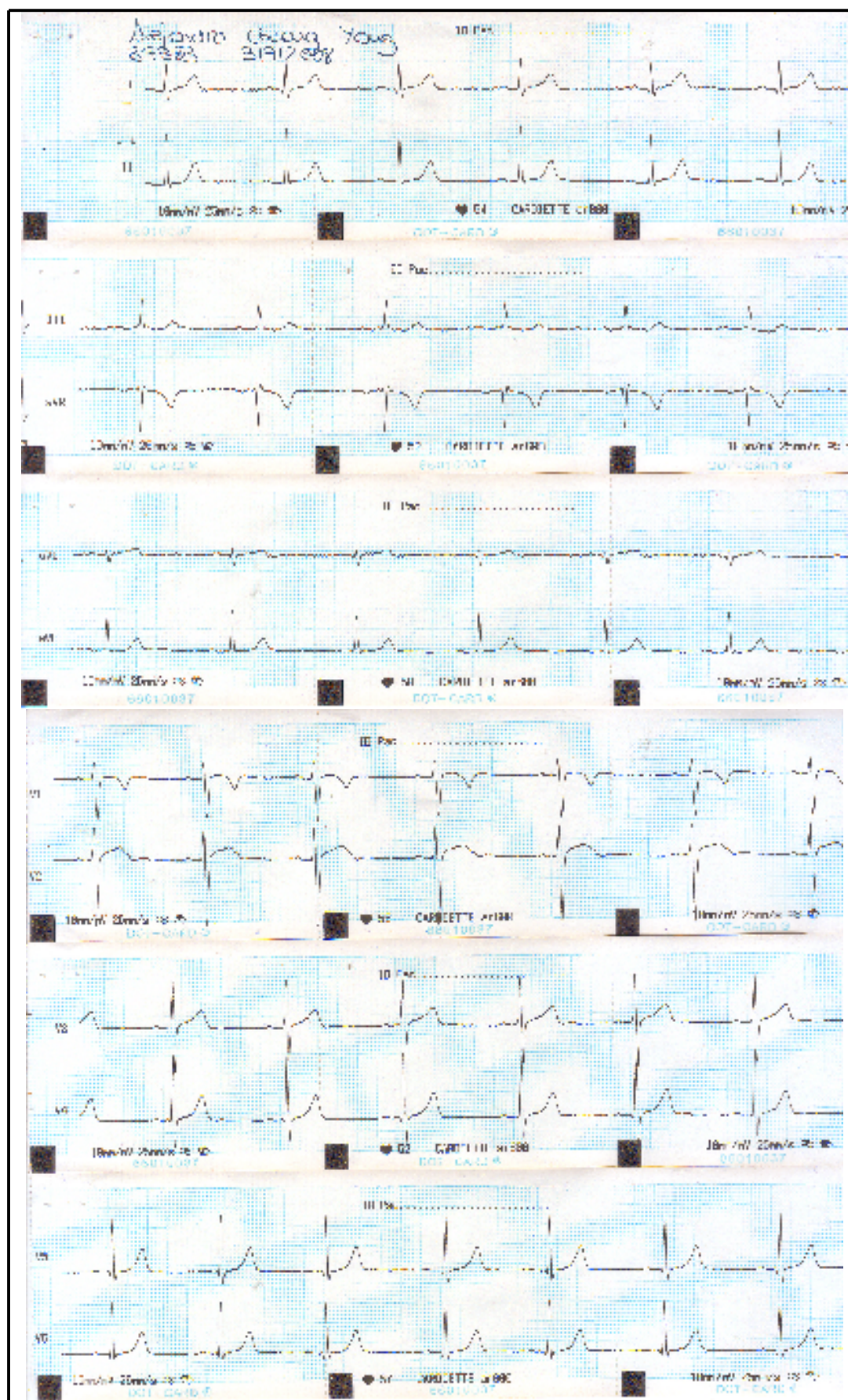


Fig. 3 Ejemplo de un ECG tomado en la actualidad.

LA DETECCIÓN de un latido implicará la detección del complejo QRS puesto que éste último es la forma de onda que más fácilmente se distingue dentro del ECG. A partir de los intervalos de tiempo entre dos ondas singulares del ECG, los cardiólogos pueden diagnosticar el estado del corazón y prevenir posibles anomalías. Sin embargo, la precisión en estas medidas de tiempo es clave para que los médicos puedan llevar a cabo decisiones correctas. El diseño del detector QRS es muy importante ya que un rendimiento pobre de éste puede tener mucha influencia en etapas posteriores y, de este modo, limitar el rendimiento del conjunto del sistema. Además, los latidos que permanecen sin detectarse constituirán un error más grave que las falsas detecciones puesto que éstos últimos son más fáciles de corregir que los primeros. En consecuencia, es tarea de los técnicos eliminar todos los posibles ruidos que puedan presentar dichas señales con el objeto de entregar a los expertos del corazón unas medidas lo más exactas posibles. Encima, el detector deberá ser capaz de detectar un gran número de morfologías QRS diferentes de forma que pueda ser clínicamente útil y capaz de seguir cambios súbitos o graduales de la morfología del QRS.

La estructura general de un detector QRS puede describirse como la del diagrama de bloques de la figura 4.

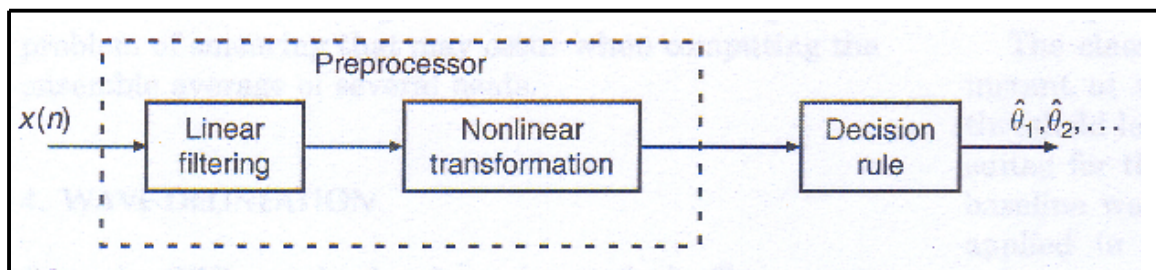


Fig. 4 Diagrama de bloques de un detector QRS³.

Dentro de esta estructura de detector, la principal función del preprocesado es la de realzar el complejo QRS al mismo tiempo que se suprimen tanto ruido como artefactos. La fase de preprocesado se implementa normalmente como un filtro lineal seguido de una transformación no lineal. Así pues, el filtro lineal deberá tener carácter paso banda para que el principal contenido espectral del complejo QRS se conserve (ver figura 5). El objetivo será mejorar la SNR para, de esta manera, incrementar el rendimiento del detector. Por su parte, la transformación no lineal realzará el complejo QRS con respecto al ruido de fondo a la vez que tratará de asemejar cada complejo QRS a un pico positivo único para una detección por umbral de amplitud más eficiente.

A la salida del preprocesador, el detector se encargará de decidir si existe un complejo QRS o no. La norma de decisión puede ser implementada como un simple procedimiento de umbral de amplitud aunque normalmente se suele imponer otras condiciones con el fin de combatir mejor el ruido. El umbral suele estar adaptado a las amplitudes más recientes de la onda para que los cambios graduales de amplitud puedan ser seguidos.

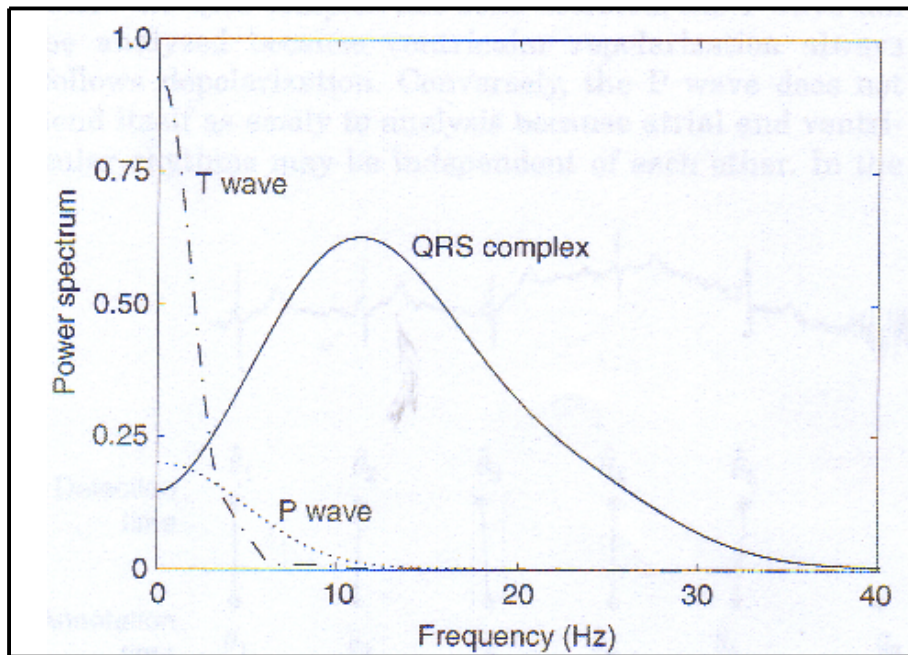


Fig. 5 Espectro de potencia de la onda P, del complejo QRS y de la onda T³.

CON EL PASO DE LOS AÑOS, se ha realizado numerosos esfuerzos para desarrollar nuevas formas de medida del ECG. Existen ciertas limitaciones del ECG clásico en cuanto a la capacidad de detectar ciertas cardiopatías o diferenciar unas de otras que han llevado al planteamiento de métodos alternativos. Uno de ellos sería el procedimiento de la vectorcardiografía, uno de cuyos principales beneficios es el de la utilización de un número mínimo de derivaciones para la medida de la electrofisiología cardíaca.

Los potenciales asociados a la actividad cardíaca generan unas corrientes iónicas que, a su vez, circularán por el tórax produciendo una distribución de potenciales superficiales. Dicha distribución es similar, en primera aproximación, a la debida a un dipolo de corriente situado en el centro eléctrico del corazón. La dirección y magnitud de dicho dipolo va cambiando a lo largo del ciclo cardíaco y su momento dipolar constituye el denominado vector cardíaco. En la vectorcardiografía, la actividad

eléctrica del corazón se modela como un vector de un dipolo que realiza bucles en un espacio tridimensional (figura 6).

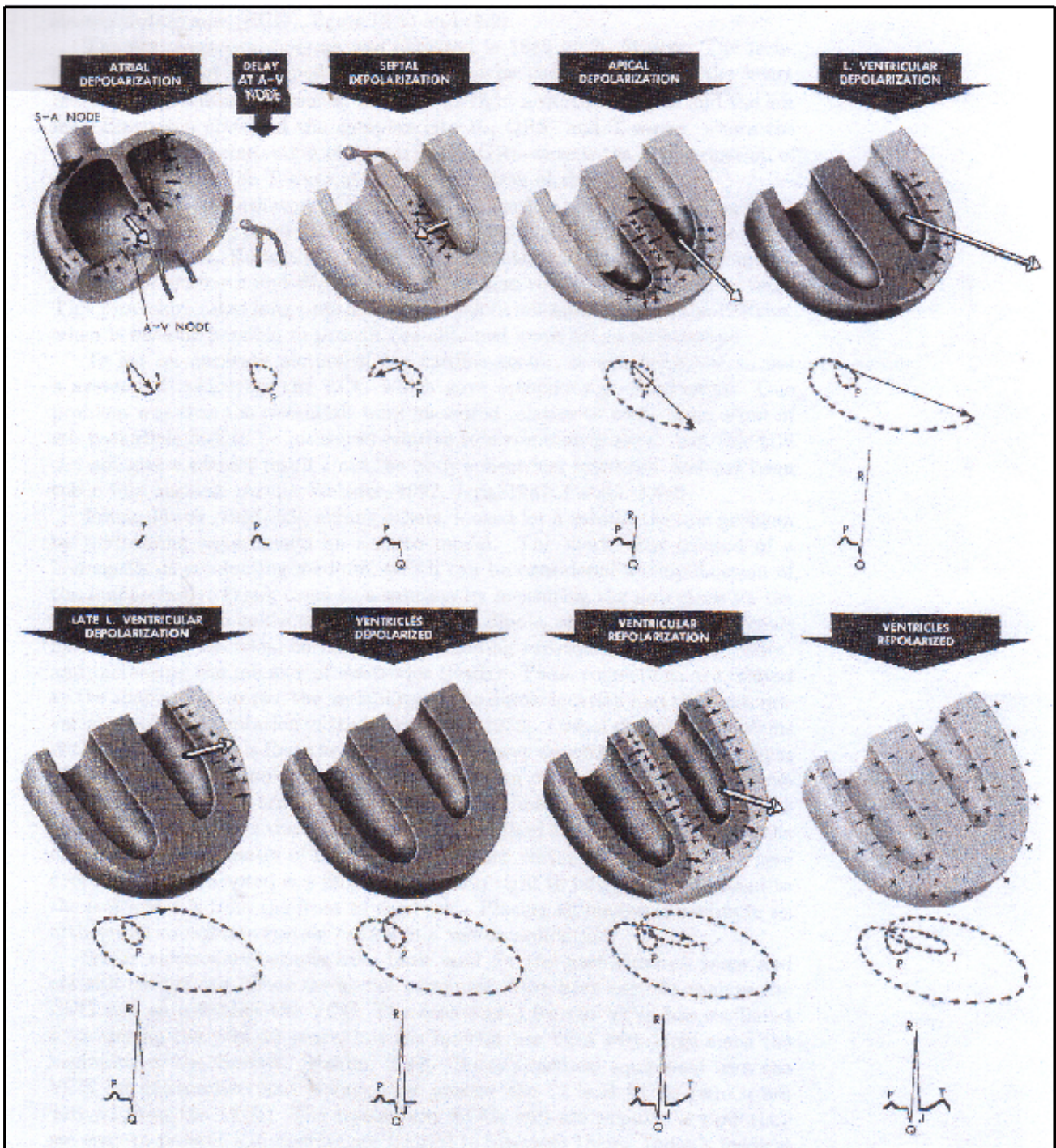


Fig. 6 El vector dipolo siguiendo un bucle durante un latido de corazón².

Así pues, el vectorcardiograma (VCG) consiste en la reconstrucción del vector cardíaco a partir de sus proyecciones en tres direcciones ortogonales X, Y, Z. La

información que se presenta es la proyección del extremo del vector cardíaco en los planos sagital, frontal y transversal (figura 7).

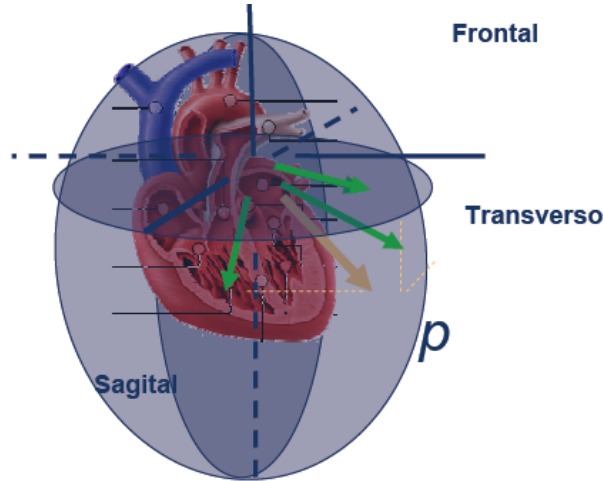


Fig. 7 Planos sagital, frontal y transversal sobre los que se proyecta el vector cardíaco⁵.

Si se conoce el comportamiento de cada componente en cada instante $\{x(t), y(y), z(y)\}$, entonces se podrá reconstruir de forma paramétrica el comportamiento del vector durante su recorrido (figura 8). Cada gráfica corresponde a la posición relativa y dirección del vector en cada momento respecto al punto de medición (se acerca o se aleja).

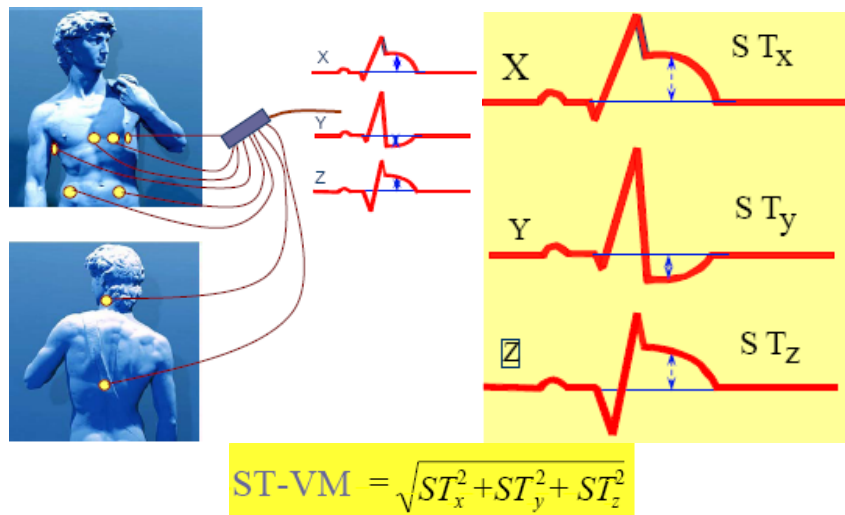


Fig. 8 Reconstrucción del vector magnitud a partir de las distintas derivaciones ortogonales⁵.

De este modo, un latido de la señal VCG se representa como una estructura de bucle consistente en una secuencia de coordenadas (X, Y, Z). En la siguiente figura se puede apreciar la representación gráfica de dicha secuencia:

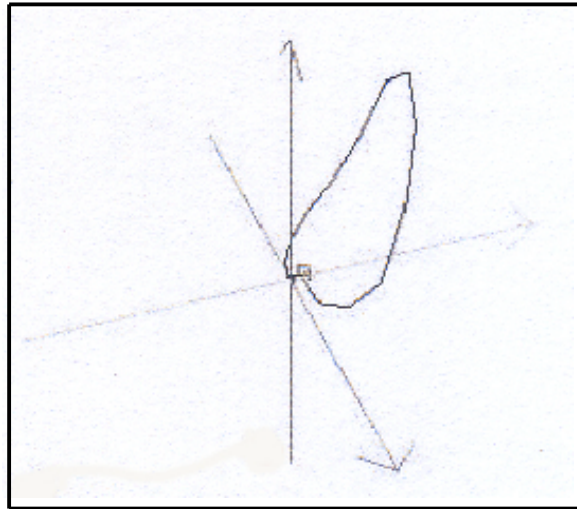


Fig. 9 Bucle de un paciente².

POR OTRA PARTE, un parámetro relevante a considerar en un sistema cardiovascular es la variabilidad de ritmo cardíaco (VRC). El VRC refleja las fluctuaciones que tienen lugar latido a latido en el ritmo cardíaco con respecto a su valor medio. Dichas fluctuaciones se derivan de la secuencia de intervalos de tiempo instantáneos entre latidos o serie RR. Su estudio es importante ya que permite una mejor comprensión de muchas características fisiológicas relacionadas con la actividad autónoma y el control cardiovascular. El análisis del VRC tiene una amplia variedad de aplicaciones clínicas y de diagnóstico como, por ejemplo, detección temprana de hipertensión, determinación de la salud de un feto, detección de las fases del sueño, detección de trombósis realizadas con éxito en pacientes con infartos de miocardio, evaluación de riesgos en pacientes que han sufrido infartos de miocardio o que padecen diabetes, etc. Además, permite el desarrollo de nuevos métodos para el análisis de fluctuaciones bajo condiciones regulares o dinámicamente irregulares (como un cuerpo sometido a un esfuerzo físico).

La señal del ritmo cardíaco latido a latido tiene dos componentes principales que van oscilando. La primera de ellas, normalmente identificada como pico de baja frecuencia, fluctúa a un periodo de aproximadamente 10 s. En el dominio frecuencial aparece normalmente como un pico espectral dentro del margen de frecuencias 0.05 y 0.18 Hz. La segunda componente oscilante refleja la modulación de la señal de ritmo

cardíaco instantánea causada por la respiración. En condiciones normales, la frecuencia de esta componente y su ancho de banda corresponden con el ritmo de respiración y su ancho espectral. Así pues, en el dominio frecuencial la componente aparecerá entre las frecuencias de 0.2 y 0.4 Hz (bajo la hipótesis de que se trata de un adulto sano) y, en consecuencia, se la conoce como pico de alta frecuencia. En la figura 10 tenemos una representación de los espectros de la variabilidad del ritmo cardíaco:

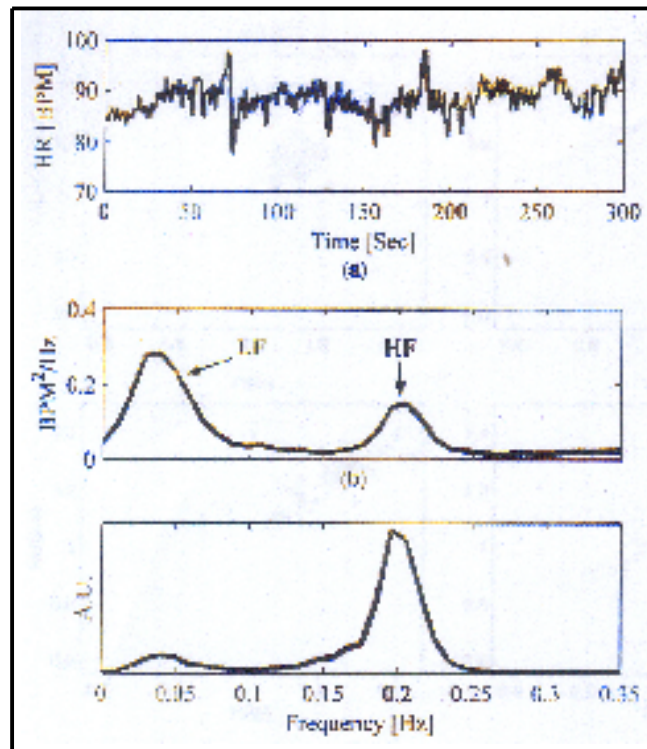


Fig. 10 Ejemplo de un registro de 300 s de duración de un ritmo cardíaco (a) y su correspondiente densidad de potencia espectral (b). (c) enseña la densidad espectral de potencia correspondiente a la señal respiratoria⁶.

LAS FLUCTUACIONES de potencia que sufren el pico de alta y de baja frecuencia están estrictamente relacionadas con la actividad de las terminaciones nerviosas del simpático y del parasimpático del sistema nervioso autónomo. En consecuencia, los cambios en la variabilidad del ritmo cardíaco pueden ser consecuencia tanto de factores no estacionarios del sistema cardiovascular (por ejemplo, cambios bruscos en la posición del cuerpo), como de la propia constitución del sujeto bajo estudio así como de la acción que desarrolla el propio sistema nervioso autónomo (por ejemplo, en el caso de la respiración).

Además, la variabilidad de ritmo cardíaco puede venir dada simplemente por el hecho de que el individuo estudiado tiene unas características fisiológicas diferentes del

estándar observado (adulto sano). Por ejemplo, el caso de la figura 11 nos muestra un estudio del ritmo cardíaco de un feto donde podemos ver la dependencia temporal de las componentes espectrales del VRC y la diferencia de morfología con respecto a la estándar.

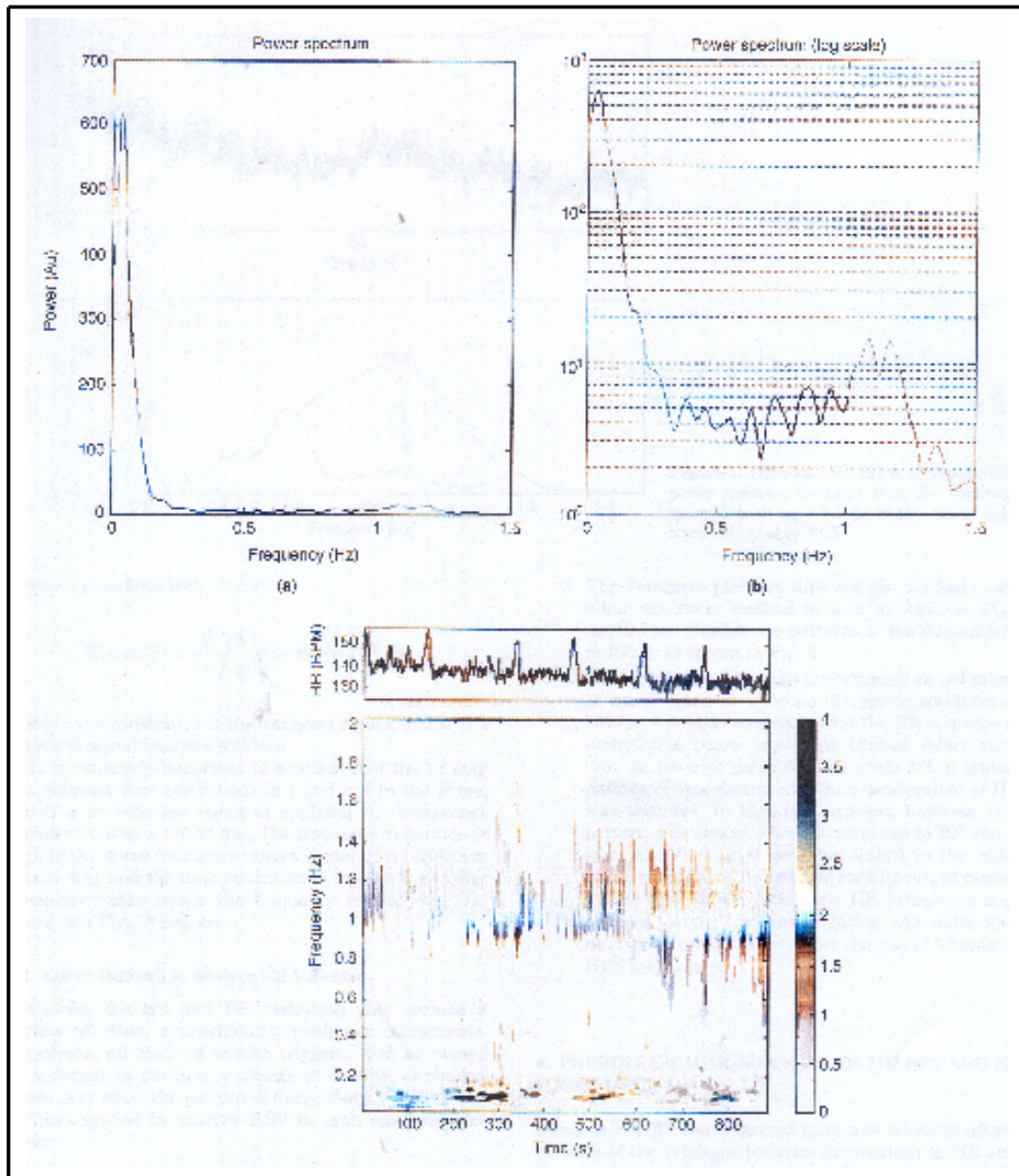


Fig. 11 Análisis del ritmo cardíaco de un feto de 38 semanas de gestación. (a) densidad espectral de potencia donde el rango de baja frecuencia (menor que 0.2 Hz) es dominante. (b) diagrama en escala logarítmica del ritmo cardíaco. (c) análisis del registro de ritmo cardíaco en tiempo-frecuencia⁶.

Un caso relevante sería el que se da en los cambios repentinos de la posición del cuerpo que también pueden afectar a la morfología del ECG. Tal y como se aprecia en la figura 12, los latidos pueden ser confundidos, en detección, por isquemias de

miocardio. Al mismo tiempo, cambios bruscos de postura del cuerpo de posición supina a estar de pie afectan a todos los mecanismos de control relacionados con el ritmo cardíaco y pueden producir grandes cambios en frecuencia y en potencia durante y después del movimiento del cuerpo.

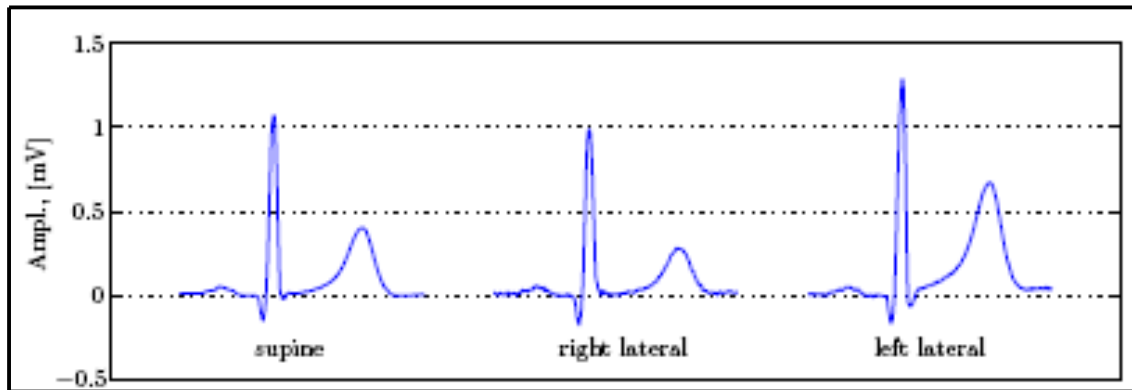


Fig. 12 Morfologías de latidos para tres posiciones medidas de un mismo individuo⁷.

IGUALMENTE, podemos encontrarnos con cambios en la morfología del ECG debidas a la acción del sistema nervioso autónomo. Una situación típica sería la de la respiración, que genera cambios en la morfología del ECG que consisten básicamente en pequeñas variaciones en la orientación espacial de los ejes que definen el vector electrocardiograma y que son coherentes con el posicionamiento del corazón.

Así pues, durante el ciclo respiratorio, los desplazamientos relativos entre el torso y el corazón provocan una rotación del eje eléctrico cardíaco, que afecta a la morfología del QRS. La variación angular que se puede producir sobre cada derivación puede llegar hasta los 5 grados⁴. Los análisis espectrales de la señal de respiración muestran un pico a la frecuencia de la respiración al que a veces acompañan algunos armónicos. La señal de respiración se deberá usar todo lo posible con el objeto de definir de la forma más precisa los posibles rangos del pico de alta frecuencia que, a su vez, puede cambiar drásticamente entre medidas de un individuo o entre individuos. La potencia del pico de alta frecuencia aumenta cuando el ritmo de respiración es reducido o cuando el volumen de aire respirado se incrementa. Por el contrario, si la frecuencia respiratoria aumenta o si el volumen respirado es reducido la potencia del pico de alta frecuencia se reducirá. Además, cabe destacar que los rangos de frecuencias descritos son las típicas para adultos de buena salud. Para sujetos enfermos, niños, fetos, etc. las frecuencias pueden ser marcadamente diferentes. Por este motivo, es muy importante

monitorizar la respiración y no sólo asumir que ocurrirá dentro de un cierto rango de frecuencias.

Como resultado de una enfermedad, la potencia de un rango de frecuencias específico puede ser aumentado (tal y como sucede con el pico de baja frecuencia en los pacientes que padecen migraña) o reducido (como en los casos de pacientes con diabetes o que han sido trasplantados de miocardio donde tanto su pico de alta como de baja frecuencia sufren cambios significativos). En el caso de pacientes trasplantados o que sufren diabetes, los individuos presentan una variabilidad de ritmo cardíaco tan débil que muchas veces las medidas tomadas estarán más ligados a cambios en la morfología del ECG del sujeto (provocadas por la respiración, por ejemplo) que a variaciones del ritmo cardíaco en sí. En consecuencia, será necesario que la serie RR sea lo más precisa posible para poder discernir entre variaciones debidas a cambios de morfología del ECG y variaciones reales del ritmo cardíaco. Para ello, antes tendremos que conocer cómo afectan estos cambios de morfología sobre el rendimiento de los detectores para poder determinar la fiabilidad de la medida final.

2. Objetivos del proyecto

El presente proyecto tiene como finalidad la verificación de la sensibilidad de ciertos detectores de QRS frente a los cambios de morfología del ECG. Para ello, se comparará los resultados obtenidos en la detección de complejos QRS en ECGs sintéticamente creados (donde se limitará el error al estrictamente debido a cambios de morfología) con los obtenidos en la detección QRS en pacientes reales.

En primer lugar, se llevará a cabo la implementación de detectores de complejos QRS a partir de algoritmos ya elaborados y extraídos de otros artículos. Los detectores que se realizarán son los siguientes: Wavelet, Genetic Algorithm, Hilbert y Zero Crossing. Además, se emplearán otros ya implementados en el proyecto de final de carrera elaborado para la UPC por Albert Monllonch: Ruha, Trahanias, Length & Energy y Hamilton-Tompkins. Se llevará a cabo una primera evaluación de dichos algoritmos sometiéndolos a la detección de un primer banco de electrocardiogramas. Todas los algoritmos se desarrollarán en MATLAB.

Seguidamente, se procederá a la creación de ECGs sintéticos que básicamente consistirán en la repetición periódica de un latido del paciente.

Posteriormente, se introducirán los cambios de morfología en dichos ECGs por medio de rotaciones graduables en las distintas derivaciones.

Para acotar la casuística del proyecto, se ha llevado a cabo un estudio para limitar los distintas situaciones e intentando simplificar al máximo. En concreto, se ha optado por escoger sólo una de las tres derivaciones del vectorcardiograma para el posterior trabajo, por la elección del mejor y peor detector para tener un mayor contraste y también por trabajar con un cambio de morfología del ECG pequeño y otro grande.

Por último, se ha iniciado la evaluación de los detectores a partir de un banco de pacientes de los que se ha extraído los ECGs sintéticos para comparar dichos resultados con los obtenidos, después, en ECGs de pacientes transplantados. De dicha confrontación se espera obtener conclusiones acerca de si el error inducido estrictamente por los cambios de morfología es negligible o no en una detección normal.

3. Descripción de los algoritmos implementados

El principal objetivo de este apartado es el de exponer la elaboración de dichos algoritmos y obtener las series RR con las que se podrá evaluar sus rendimientos globales.

La implementación de los algoritmos se ha llevado a cabo siguiendo los procedimientos pautados en artículos publicados por otros autores. En total se ha llevado a cabo la realización de cuatro detectores aunque para el estudio se emplearán otros cinco adicionales ya implementados. Los cuatro algoritmos realizados cogerán un electrocardiograma de entrada y localizarán los picos R. A la salida del detector, dispondrán un tren de deltas (donde cada delta equivaldrá a un pico R estimado). Además, indicarán a la salida el número de deltas que compone dicho tren de tal manera que podremos comprobar fácilmente el número de latidos detectados. Se ha procurado que los detectores tengan como parámetro de entrada la frecuencia de muestreo por la simple razón de que los ECGs pueden venir muestreados a frecuencias distintas. Por último, se procederá a la captura de la serie RR de cada ECG filtrado que nos permitirá profundizar en la evaluación del rendimiento de los detectores por medio de la variación del ritmo cardíaco.

Tal y como se ha apuntado en la introducción, la estructura común que seguirán dichos detectores será la de una primera fase de preprocesado en la que se procurará realzar la señal y prepararla para la segunda fase, la detección. De esta manera, se logrará mejorar la relación señal a ruido con la consecuente mejora de la calidad de la detección final.

3.1 Detector de ECGs mediante la transformada Wavelet

La implementación se ha basado en la propuesta realizada por Li en 1995⁸ que emplea las propiedades de la transformada wavelet para optimizar la detección de ondas R.

La transformada wavelet sigue la siguiente expresión:

$$W_s f(x) = f(x) * \Psi_s(x) = \frac{1}{s} \int_{-\infty}^{+\infty} f(t) \Psi\left(\frac{x-t}{s}\right) dt$$

donde s es un factor de escala y $\Psi(x)$ es un pulso wavelet básico o pulso madre.

En el detector hemos utilizado el pulso madre “Haar” (cuya forma podemos ver en la figura 13) para llevar a cabo la transformada wavelet.

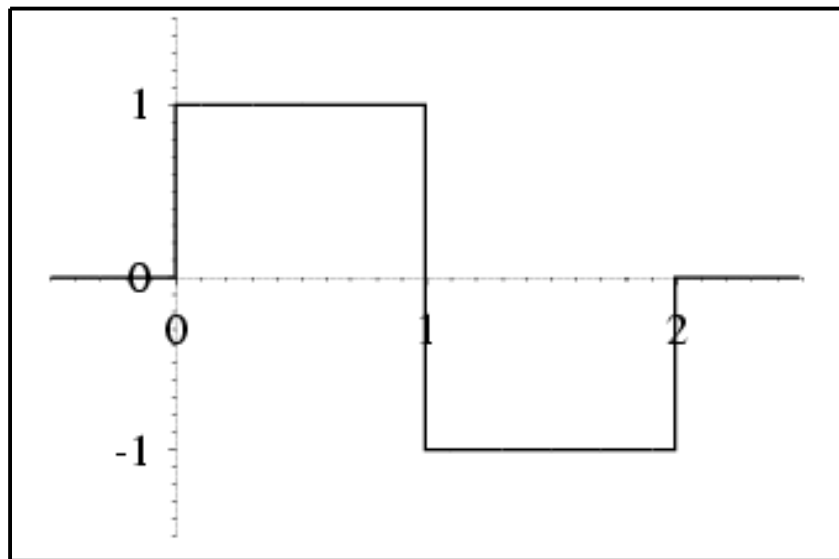


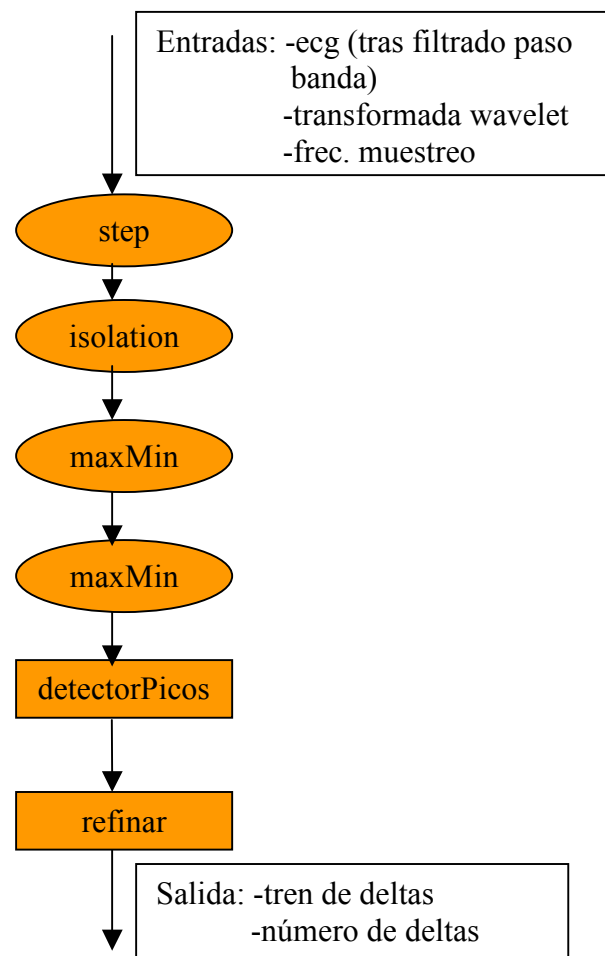
Fig. 13 Pulso madre de la wavelet Haar.

Dado que el procesamiento de señales ECG de un número de muestras muy superior a 100.000 (si la frecuencia de muestreo es de 250 Hz, esto equivaldría a ventanas de 400 s) aumentaba considerablemente el tiempo de proceso del programa se ha optado por la realización del detector wavelet por medio de una implementación por bloques. De esta forma, se subdivide el electrocardiograma en fragmentos de 100.000 muestras con el fin de optimizar el tiempo de procesamiento (para ECGs de duración mayor que 400s). Cada uno de estos bloques serán filtrados por el bloque “wl” y los trenes de deltas obtenidos

de cada filtrado se irán concatenando de forma que el tren de deltas final sea la unión de los obtenidos en todos los subbloques. Asimismo, se llevará a cabo una cuenta del número de deltas que se vayan acumulando para todos los bloques.

Antes de entrar en el bloque “wl”, la señal es filtrada paso banda por un filtro de Butterworth de orden 2 con frecuencias de corte de 10 y 30 Hz.

El bloque “wl” consta de los siguientes bloques:



Dicho bloque consta de las siguientes partes:

-step: se encarga de localizar las líneas de módulos máximos (pares máximo local y mínimo local cuya presencia en el dominio de la transformada wavelet representan una pendiente muy pronunciada en el dominio temporal de la señal; es decir, en nuestro caso sería un pico R) de las ondas R empezando desde la escala 2^4 hasta la escala 2^1 (el porqué de ello será explicado más adelante).

En la figura 14 se puede apreciar las distintas respuestas frecuenciales que las transformadas wavelets tienen en función de la escala empleada. Se trata, pues, de filtros equivalentes de los que se desprende que, al aumentar el valor de la escala, las componentes espectrales de la señal original pasarán progresivamente a las bajas frecuencias. Además, el ancho de banda del filtro también irá decreciendo. Nótese que al variar la frecuencia de muestreo también cambiará el valor de la escala de forma proporcional. Por ejemplo, si tenemos una escala de 2 a una frecuencia de muestreo de 250 Hz y pasamos a una de 1000 Hz nuestra escala deberá cambiar a 8.

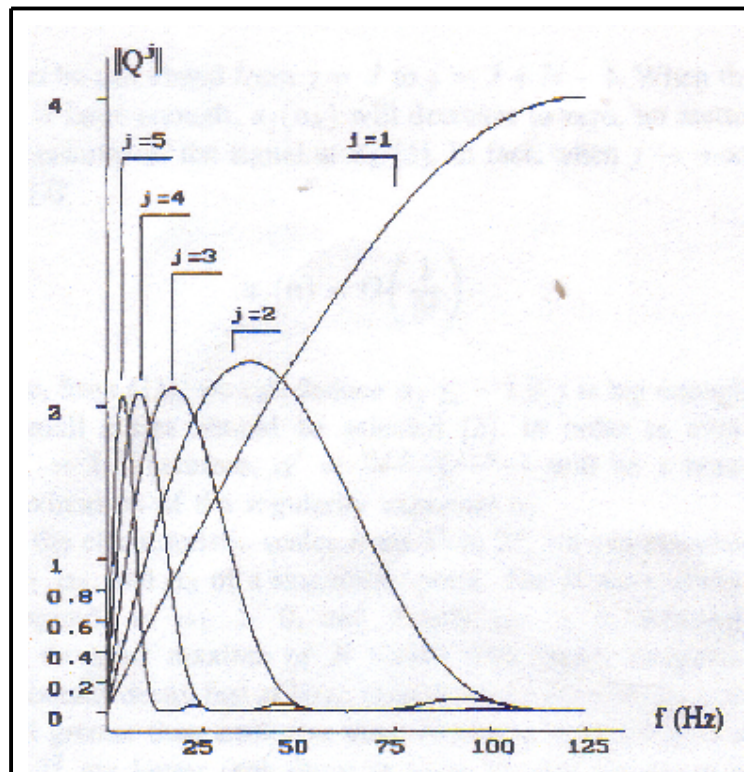


Fig. 14 Respuestas frecuenciales del módulo del filtro equivalente de la wavelet (a una $f_m=250$ Hz)⁸.

Los máximos locales del módulo de la transformada wavelet (en las diferentes escalas) serán utilizadas para detectar puntos singulares del ECG donde exista una gran variación de pendiente. Como se puede apreciar en la siguiente figura, las rampas de subida corresponden a un mínimo negativo y las rampas de bajada a un máximo positivo (en las distintas escalas). Los módulos de estos máximos o mínimos correspondientes al mismo pico se llaman líneas de módulos máximos (ver figura 15).

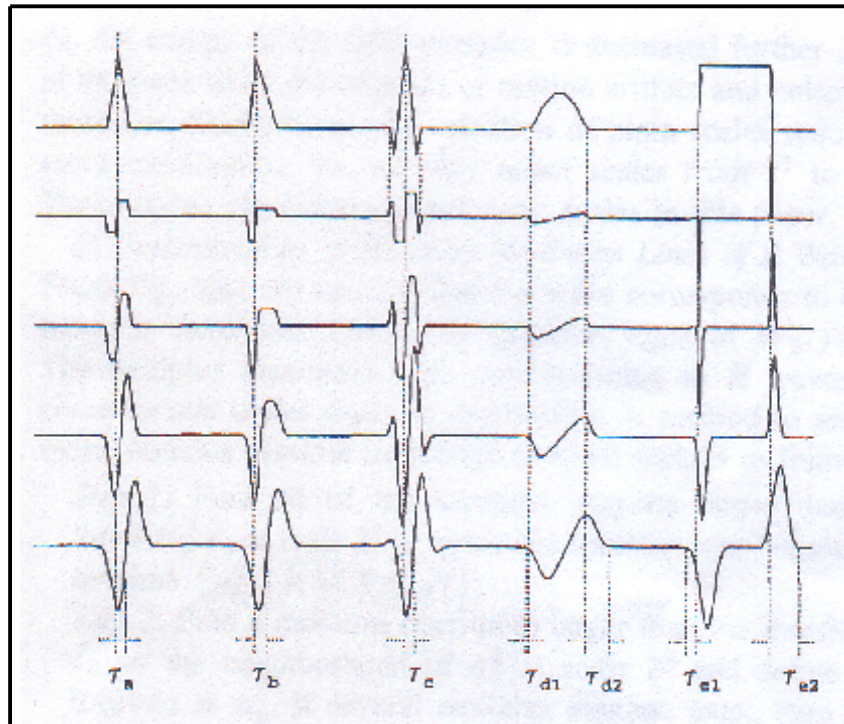


Fig. 15 Relación entre los puntos característicos de una simulación de ondas y transformadas wavelets en las distintas escalas⁸.

El objeto de la búsqueda de líneas de módulos máximos empezando por una escala mayor y disminuyéndola paso a paso es que el número de módulos máximos son mucho menores en las primeras (ya que su ancho de banda es menor) y, por lo tanto, será más restrictivo a la hora de seleccionar candidatos a picos QRS de tal forma que se pueda ahorrar tiempo de proceso (Ver figura 16). Además, el ruido de alta frecuencia decae mucho en las escalas altas de tal forma que no se generan líneas de módulos máximos o se propician muy pocas.

scale	3 db bandwidth (Hz)
$s=2^1$	62.5 ~ 125.0
$s=2^2$	18 ~ 58.5
$s=2^3$	8 ~ 27
$s=2^4$	4 ~ 13.5
$s=2^5$	2 ~ 6.5

Fig. 16 Anchos de banda a 3 dBs en las distintas escalas del filtro equivalente⁸.

En el siguiente flujograma, se puede observar el funcionamiento de la función “step” (donde $WT2^i(n)$ es la transformada en la escala 2^i -ésima de la señal, $PS2^i(n)$ son

los puntos de $WT2^i(n)$ que sobrepasan un cierto umbral E_i y N hace referencia al número de puntos candidatos a líneas de módulos máximos que a medida que avance el filtrado se irán acotando). Los umbrales E_i se han determinado de forma empírica.

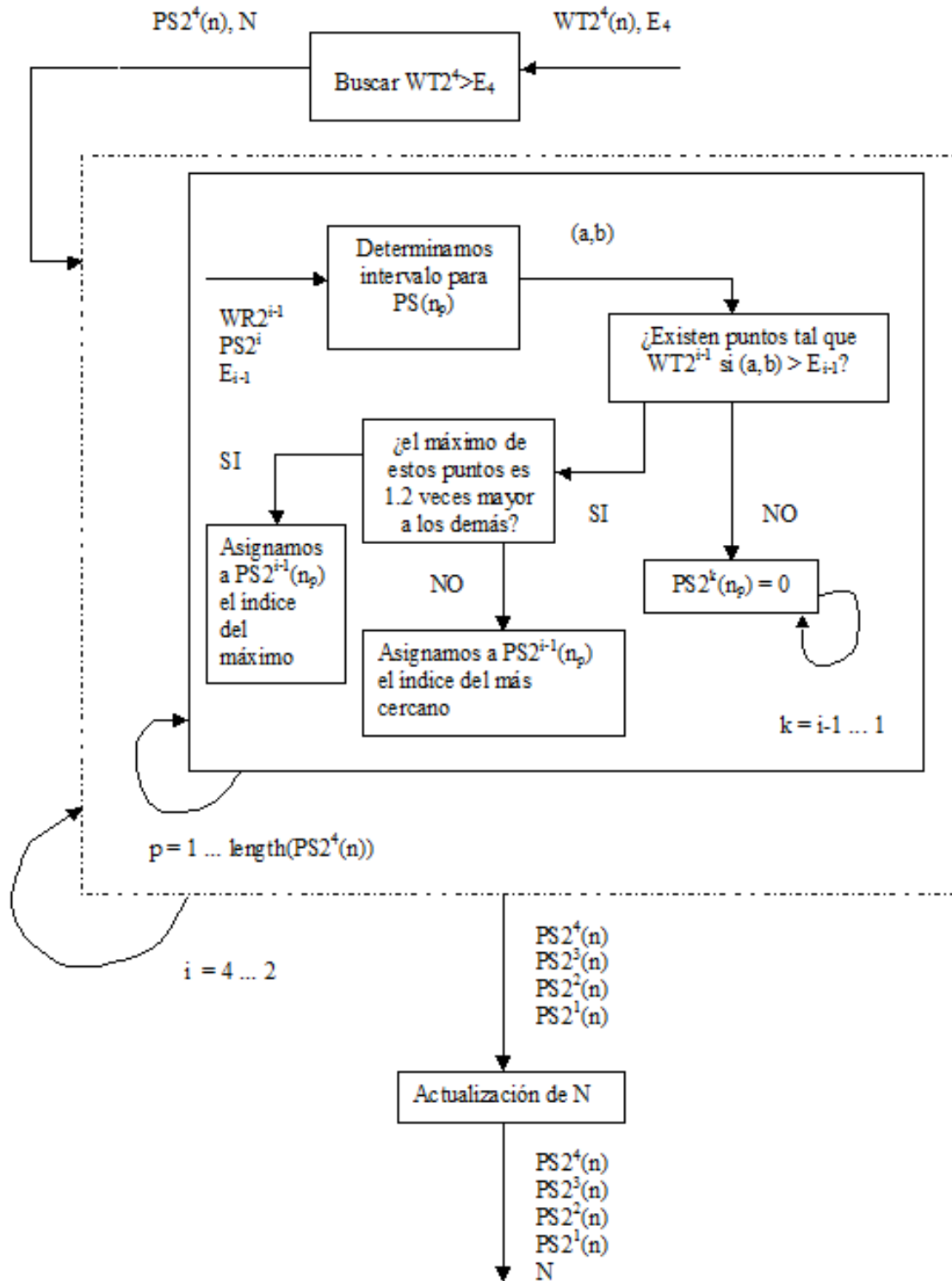


Fig. 17 Flujograma de la función "step".

Los dos siguientes módulos se encargarán de eliminar las bandas de frecuencia debidas a los artefactos móviles y al ruido de los músculos que con frecuencia se superponen al QRS complejo (y que los filtros paso banda convencionales no pueden eliminar):

-isolation: elimina las líneas de módulos máximos que están aisladas (es decir, que no correspondan a un par máximo positivo mínimo negativo asociadas a un pico en el dominio temporal) puesto que la onda R corresponde a un par máximo positivo-mínimo negativo en cada escala característica. Se ha empleado un intervalo umbral de 120 ms. Su implementación es la que sigue:

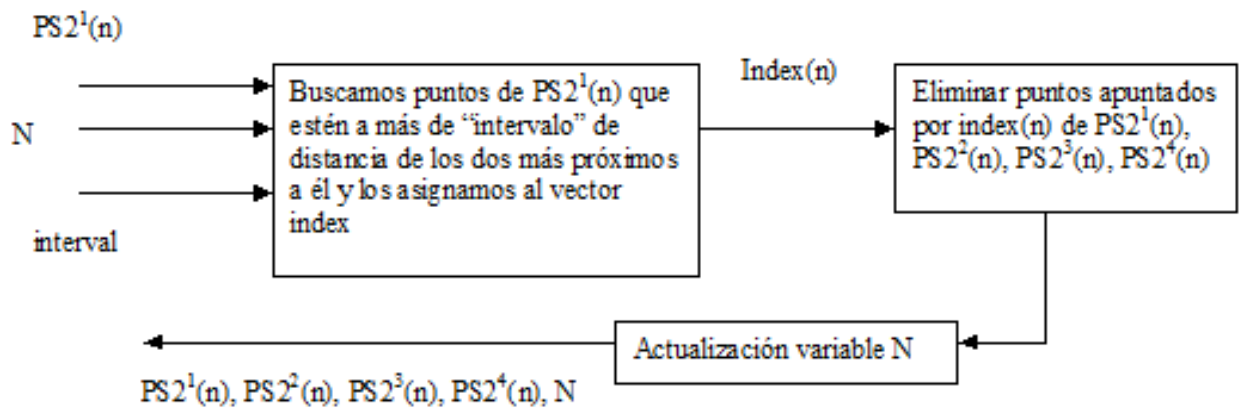


Fig. 18 Flujograma de la función "isolation".

-maxMin: elimina las líneas de módulos máximos redundantes. Normalmente, la onda R corresponde a dos líneas de módulo máximo. Sin embargo, debido a posibles ruidos en un entorno de dicha onda (hemos considerado como entorno un intervalo de 120 ms) pueden aparecer dos o más líneas de módulo máximo de las que sólo escogeremos una y, por consiguiente, descartaremos el resto.

El siguiente flujograma ilustra el funcionamiento de dicha función cuyos criterios de selección de líneas de módulo máximo se describen con detalle en el artículo aludido anteriormente. Cabe destacar que se trata de una implementación dual que permitirá filtrar aquellos casos en los que se dé un máximo y dos mínimos (opción 1) así como las situaciones en las que aparezcan dos máximos cerca de un mínimo (opción 2).

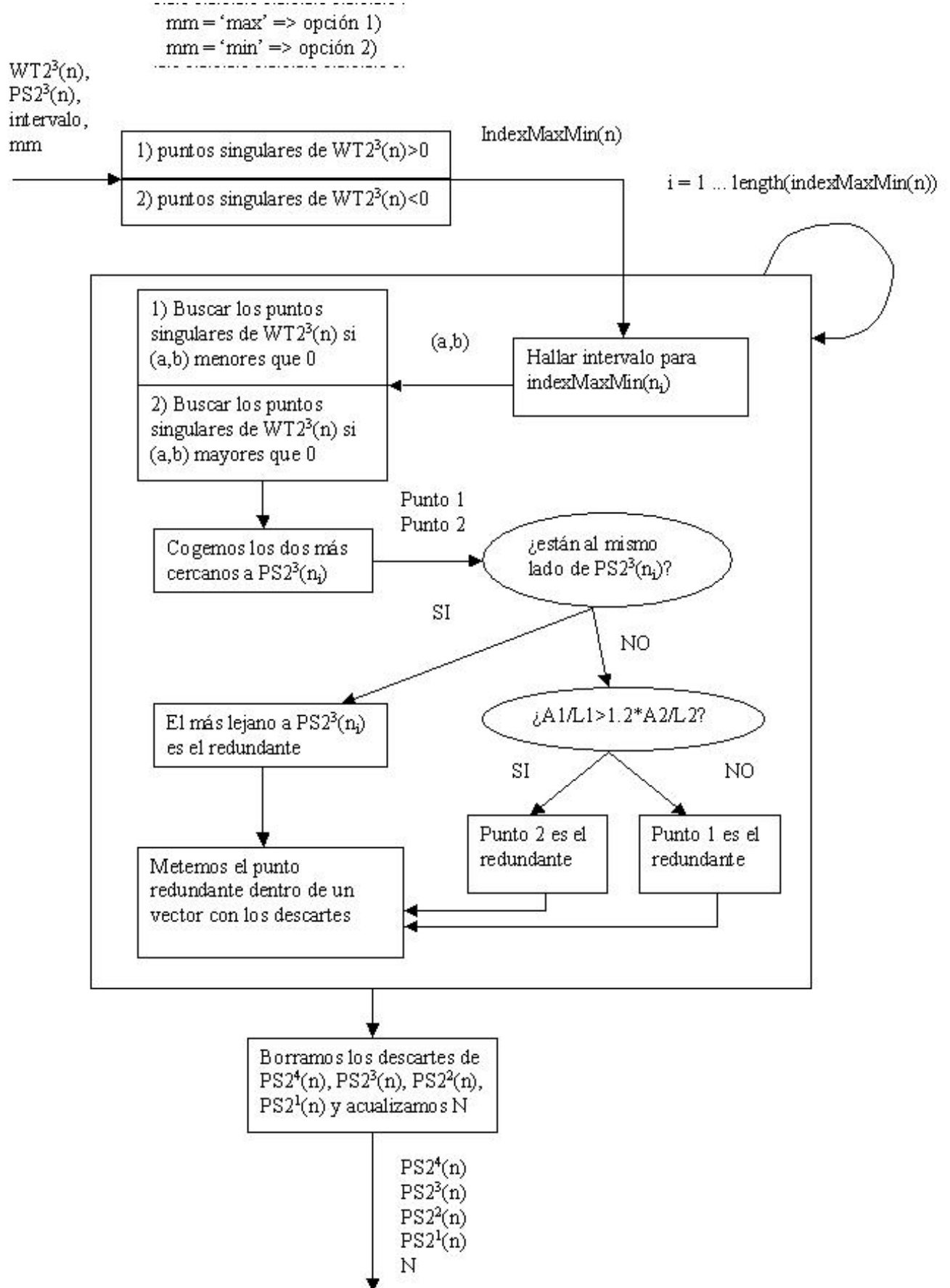


Fig. 19 Flujograma de la función “maxMin”.

La figura 20 nos presenta la variable ps21 justo antes de la entrada del detector de picos y nos indica los puntos candidatos que finalmente han sobrevivido al filtrado de todos los bloques anteriores. En la figura 21, que simplemente es una ampliación de la fig. 20, podemos ver que cada sección horizontal de la señal corresponde a un punto candidato a delta.

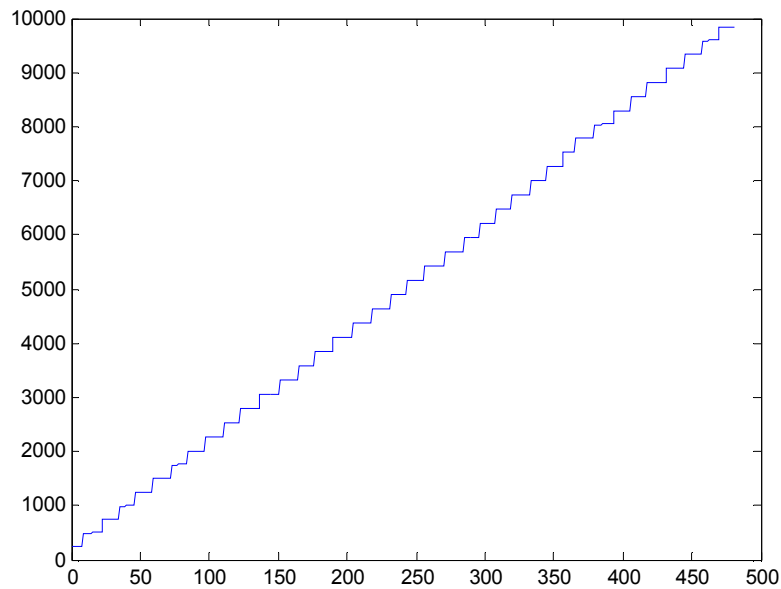


Fig. 20 Variable ps21

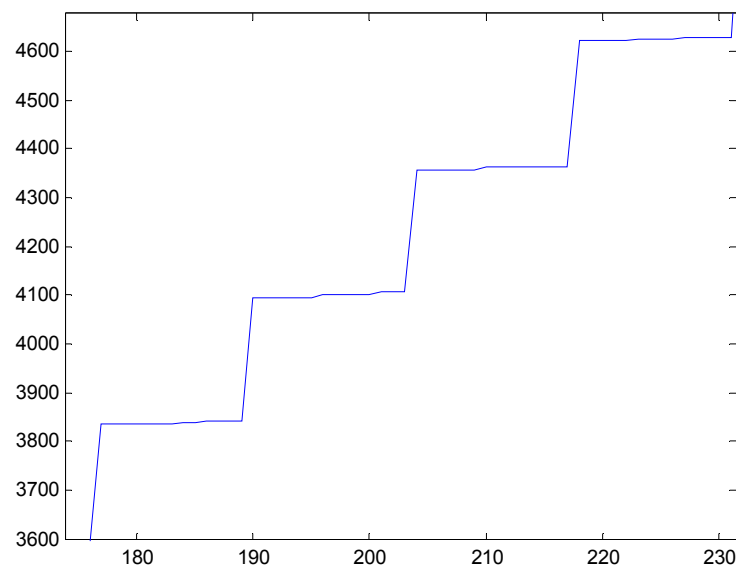


Fig. 21 Cada sección horizontal de la señal nos indican un punto candidato a delta.

-detectorPicos: hallará los picos R localizados en los pasos por cero de máximo positivo a mínimo negativo en la escala 2^1 . El siguiente flujograma nos da una idea de su funcionamiento:

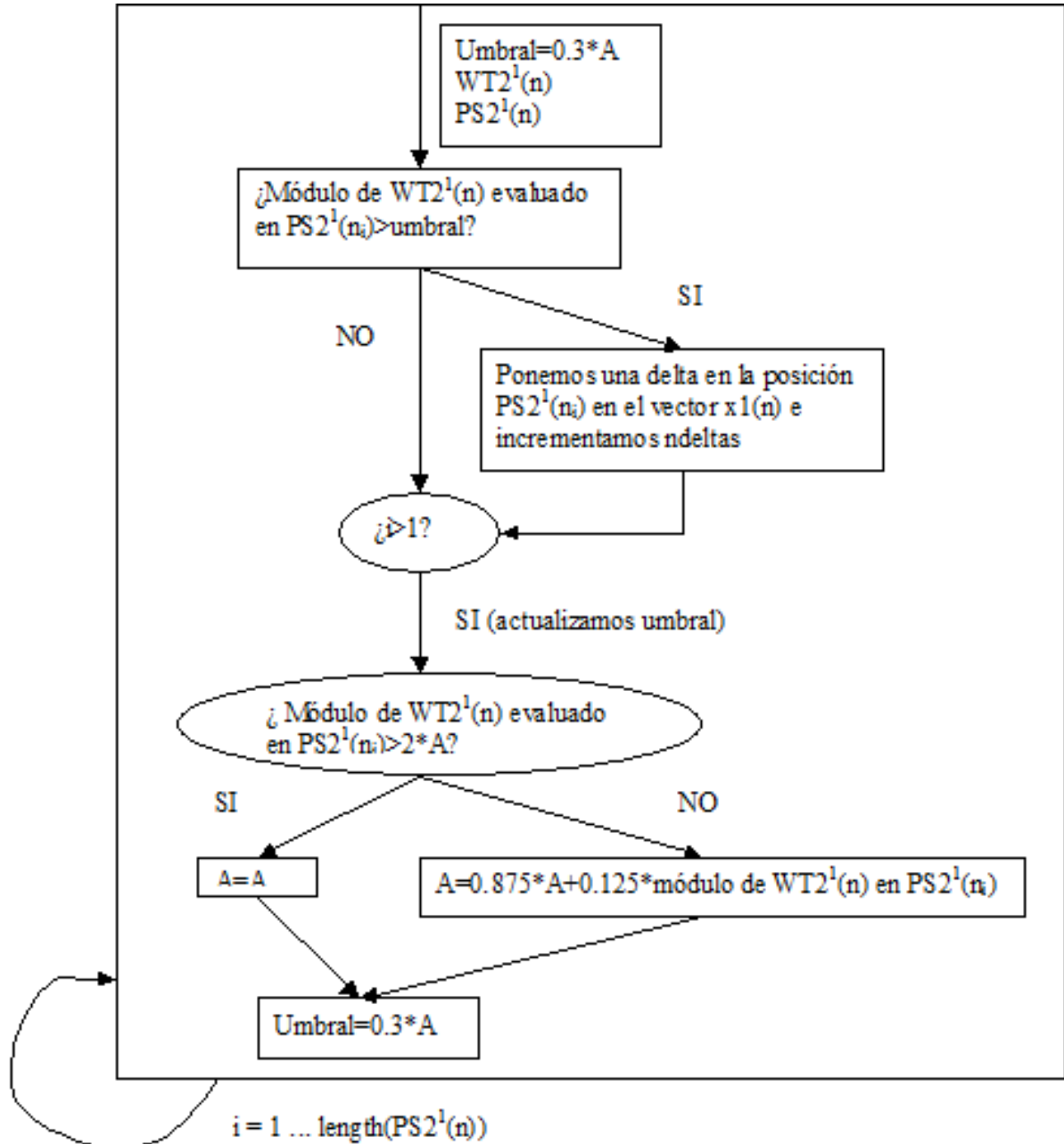


Fig. 22 Flujograma de la función “detectorPicos”.

A su salida obtendremos un tren de deltas (figura 23) donde cada delta nos indicará la situación de cada pico R (figura 24).

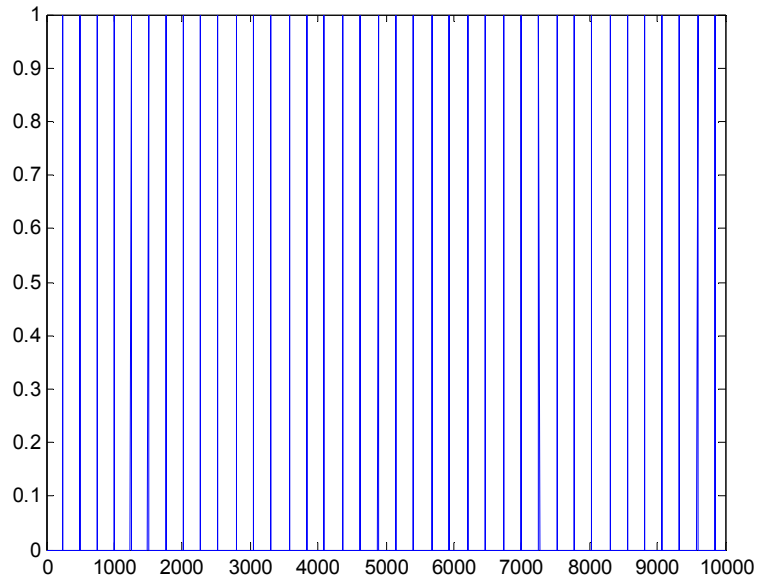


Fig. 23 Tren de deltas a la salida del filtro (nótese que en el eje X se trata de muestras)

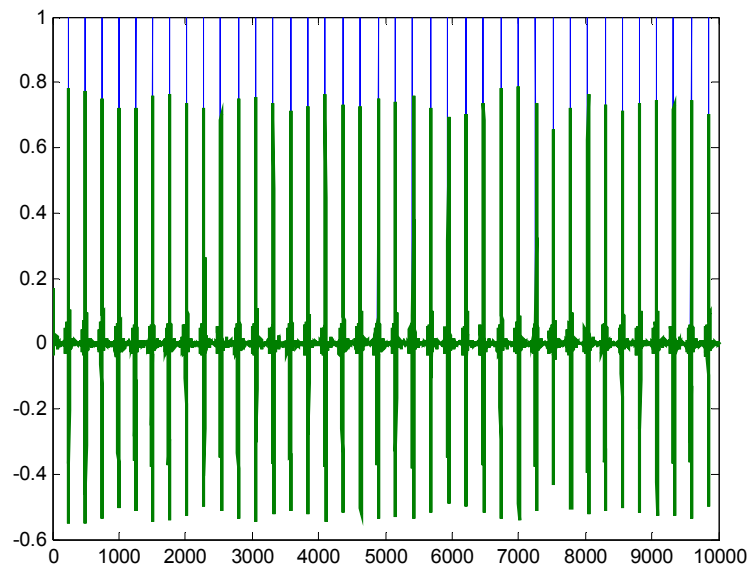


Fig. 24 Tren de deltas sobrepuesta al ECG.

-refinar: a partir del tren de deltas obtenido, se busca dentro del ECG el máximo allá donde se encuentren unos. Así pues, a estos nuevos máximos se les asignará un uno en un nuevo tren de deltas.

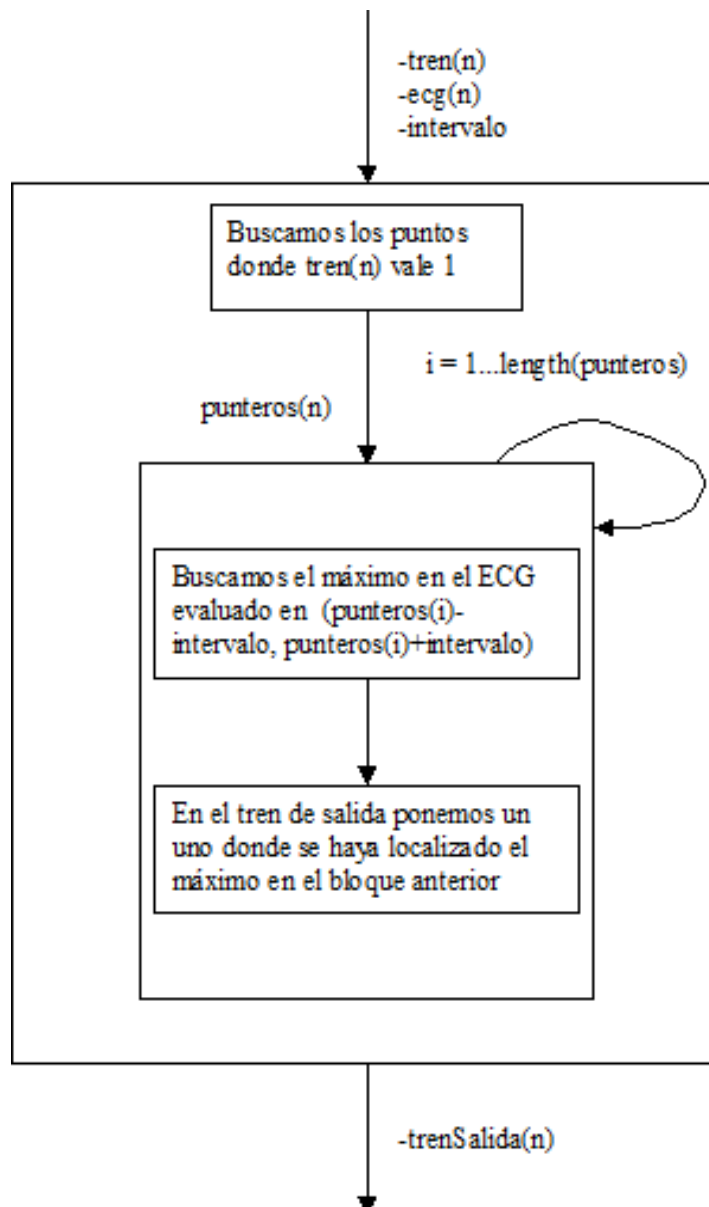


Fig. 25 Flujograma de la función “refinar”.

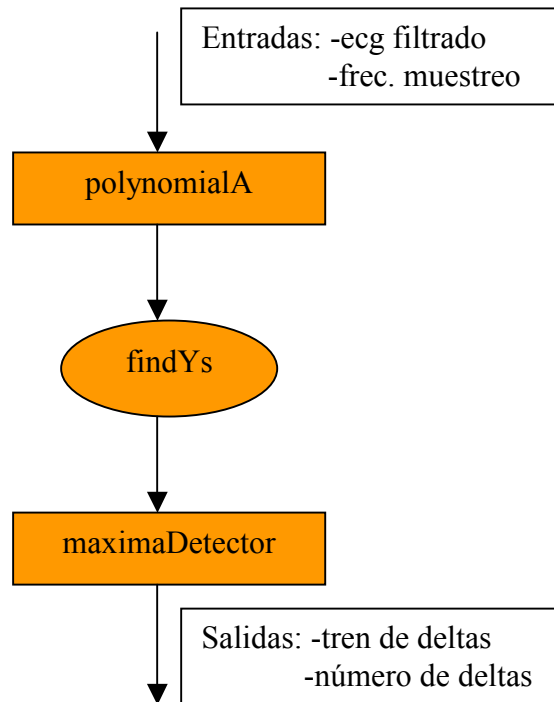
3.2 Detector de ECGs mediante algoritmos genéticos

Para la realización de este detector nos hemos basado en la propuesta de R. Poli, S. Cagnoni y G. Valli (1995)⁹.

El detector está constituido por dos módulos: el primero consiste en un filtro polinómico que realza y rectifica la señal QRS y el segundo es un detector de máximos adaptativo. Con el fin de obtener una mayor eficiencia, el filtro operará con un número pequeño de muestras en la entrada.

Antes del detector se ha llevado a cabo un preprocesado de la señal consistente en un filtrado paso banda del ECG por medio de un filtro de Butterworth con frecuencias de corte a 10 y 30 Hz.

El detector consta de los siguientes bloques:



-polynomialA: se trata de un filtro polinómico que implementa la siguiente ecuación:

$$y_i = \underbrace{\sum_{k_1=0}^M \sum_{k_2=0}^M \cdots \sum_{k_N=0}^M}_{\sum k_j \leq M} a_{k_1 k_2 \cdots k_N} x_{i-d_1}^{k_1} x_{i-d_2}^{k_2} \cdots x_{i-d_N}^{k_N}$$

donde y_i es la señal de salida en el instante i , M es el orden del algoritmo polinómico, N es el número de muestras de entrada que se cogen en cada instante y las d_j son los retardos con respecto al instante i .

En nuestro caso, hemos tomado los valores obtenidos empíricamente por los autores mencionados anteriormente. Dichos valores se corresponden a un filtro polinómico de orden 1 que va tomando entradas de 10 muestras. Sus coeficientes son los que siguen:

$a_{1000000000}$	-7.13024
$a_{0100000000}$	-7.94744
$a_{0010000000}$	3.04609
$a_{0001000000}$	4.77475
$a_{0000100000}$	9.22591
$a_{0000010000}$	7.31803
$a_{0000001000}$	9.37990
$a_{0000000100}$	-8.26923
$a_{0000000010}$	-9.57643
$a_{0000000001}$	-0.62212

Notar que estos coeficientes extraídos del artículo mencionado anteriormente son para un filtrado de ECGs muestreados a 120 Hz y, por lo tanto, se ha tenido que adaptar los retardos de la ecuación polinómica para hacerlos compatibles con la frecuencia de muestreo de los bancos de muestras empleados. La transformación llevada a cabo sobre la señal sigue el siguiente proceso:

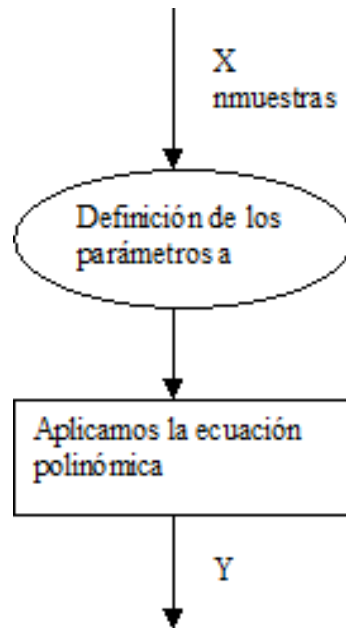


Fig. 26 Flujograma de la función “polynomialA”.

En las figuras 27 y 28 podemos comparar señales de antes y después del filtrado polinómico.

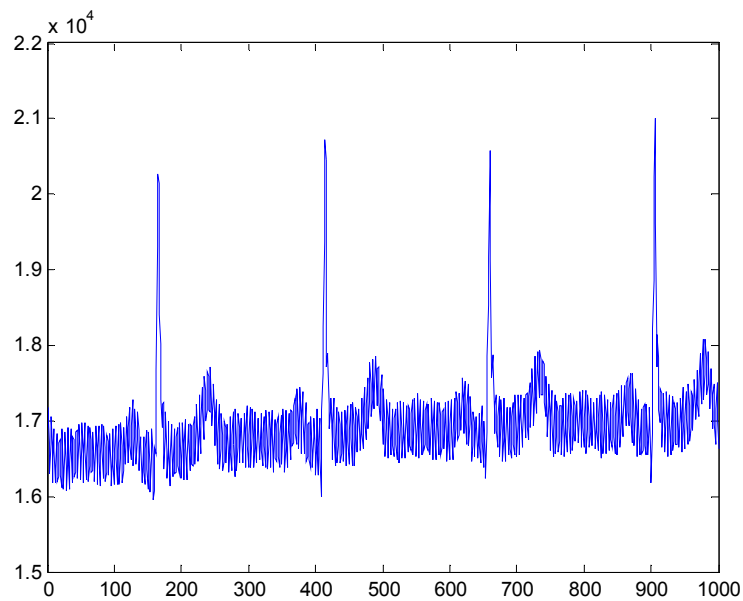


Fig. 27 Señal ECG previa al filtrado polinómico.

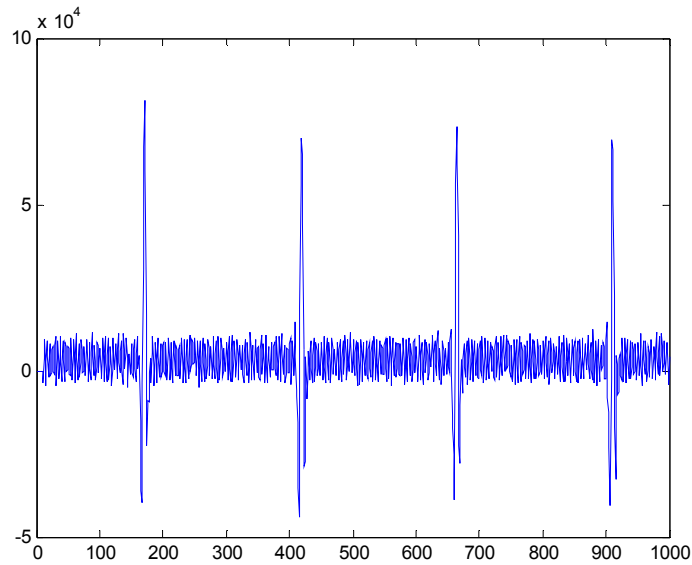


Fig. 28 Señal ECG una vez filtrada.

-findYs: dicha función busca los parámetros iniciales para realizar la detección de máximos. Así pues, estimará los valores Y_{start} , Y_{min} , Y_{max} a partir de una media de los valores de los máximos locales hallados en la señal Y ($Y_{start}=0.7*m$, $Y_{max}=0.9*m$, $Y_{min}=0.5*m$, siendo “ m ” la media de los máximos locales del ECG. Los coeficientes han sido hallados de forma empírica). Su funcionamiento es el que sigue:

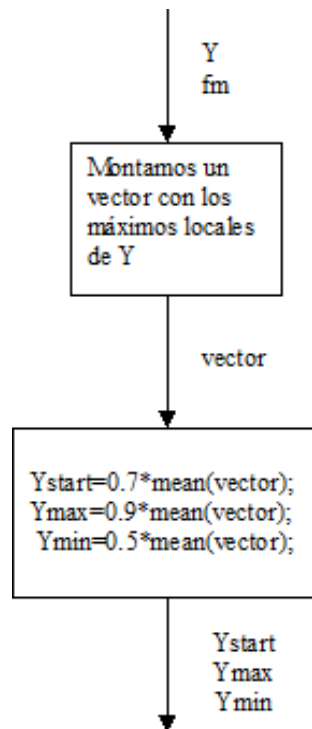


Fig. 29 Flujoograma de la función “findYs”.

-maximaDetector: a la salida del filtro anterior, detecta los máximos cuya amplitud supere un cierto umbral Y y que no decaiga dentro de un periodo de C muestras tras el último QRS detectado. Con el fin del combatir la variabilidad de la señal ECG, los autores del artículo han llevado a cabo un esquema adaptativo para Y . Este umbral adaptativo posibilitará que el proceso de detección pueda empezar a pesar de la presencia de señales ECG con amplitudes muy pequeñas. Tras cada detección, el umbral se aproximará al valor objetivo determinado y que está representado por un porcentaje de la amplitud del último pico detectado.

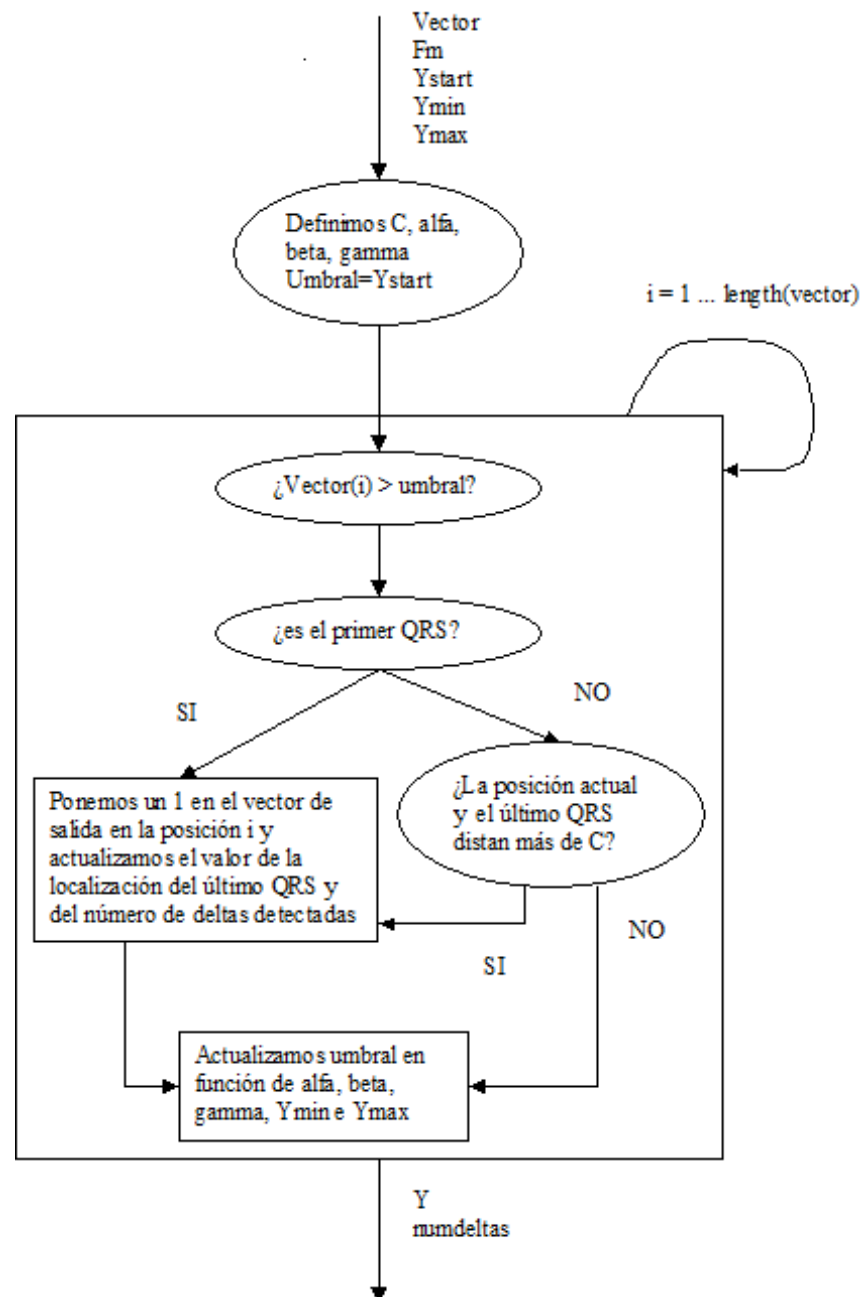


Fig. 30 Flujograma de la función “maximaDetector”.

La actualización del umbral sigue el siguiente algoritmo:

$$Y_0 = Y_{\text{start}}$$

$$Y_i = g[(1-\alpha) Y_{i-1} - \beta_{zi-1}(Y_{i-1} - \gamma_{yi-1})]$$

donde $g(x) = \max\{Y_{\min}, \min[Y_{\max}, x]\}$. Los valores de α , β , γ son los que aparecen en el artículo.

A la salida obtendremos un tren de deltas como el de la figura 31.

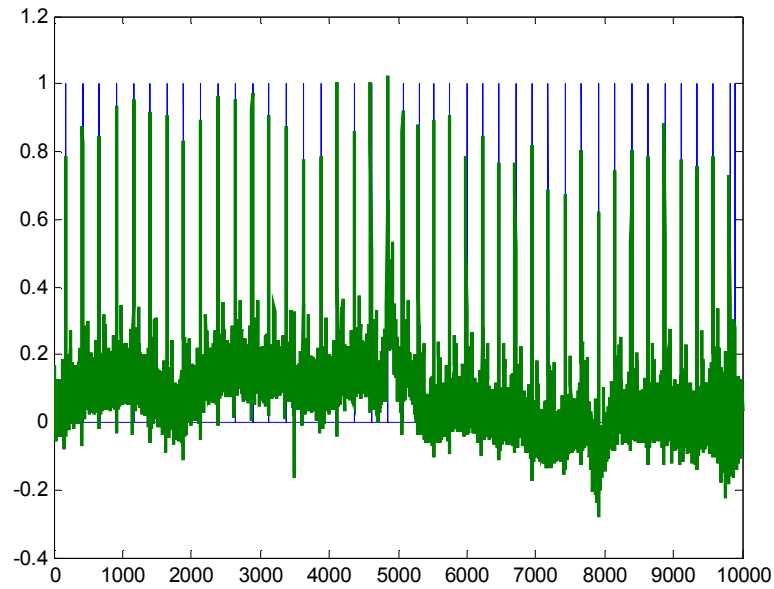
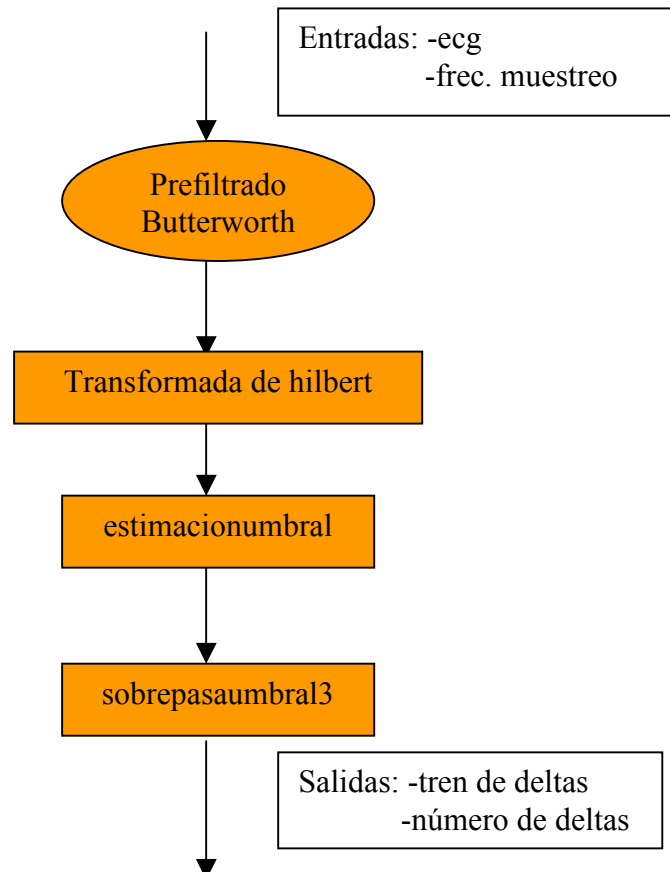


Fig. 31 Tren de deltas de la salida sobrepuesta al ECG de entrada.

3.3 Detector de ECGs mediante la transformada de Hilbert

Nuestra realización se basa en la propuesta que aparece en el artículo de Köhler, Hennig y Orglmeister (2002)¹⁰.

Dicho detector hace uso de la transformada de Hilbert para mejorar la detección de los complejos QRS. Está compuesto por los siguientes módulos:



-prefiltrado: elimina el ruido de fuera de la banda de interés por medio de un filtro Butterworth de orden 2 y frecuencias de corte 10 y 30 Hz.

-transformada de Hilbert: aplica la transformada de Hilbert a la señal ECG.

$$x_H(t) = \frac{1}{\pi} \int_{-\infty}^{\infty} \frac{x(\tau)}{t - \tau} d\tau = x(t) * \frac{1}{\pi t}$$

La función “hilbert” que hemos utilizado, que pertenecía a la librería de MATLAB disponible, usa la FFT para pasar las muestras a dominio frecuencial discreto

donde realiza las operaciones pertinentes y, posteriormente, por medio de la FFT inversa devolverá la señal al dominio temporal.

En la figura 32 podemos ver la señal antes de la transformada de Hilbert y en la 33 una vez transformada. Nótese que lo que se emplea en la transformada de Hilbert (que es una señal compleja) es su módulo.

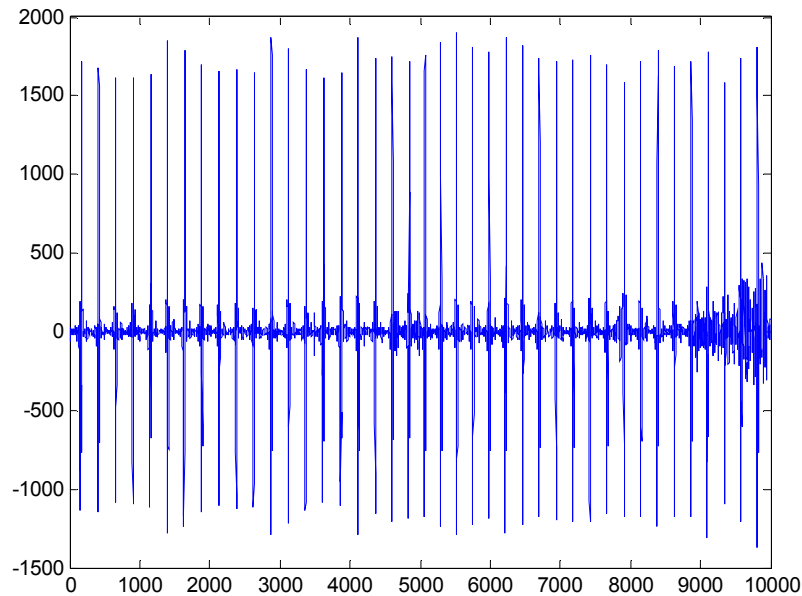


Fig. 32 ECG previo transformada de Hilbert.

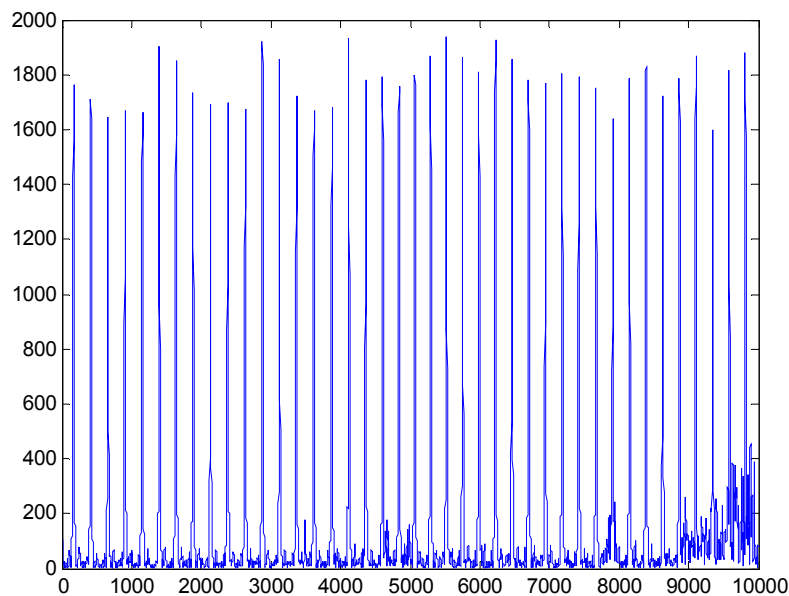


Fig. 33 ECG tras la transformada de Hilbert.

-estimaciónumbral: se trata de una función que obtiene un vector de umbrales estimados a través de un algoritmo adaptativo¹³. A continuación se representa su diagrama de bloques:

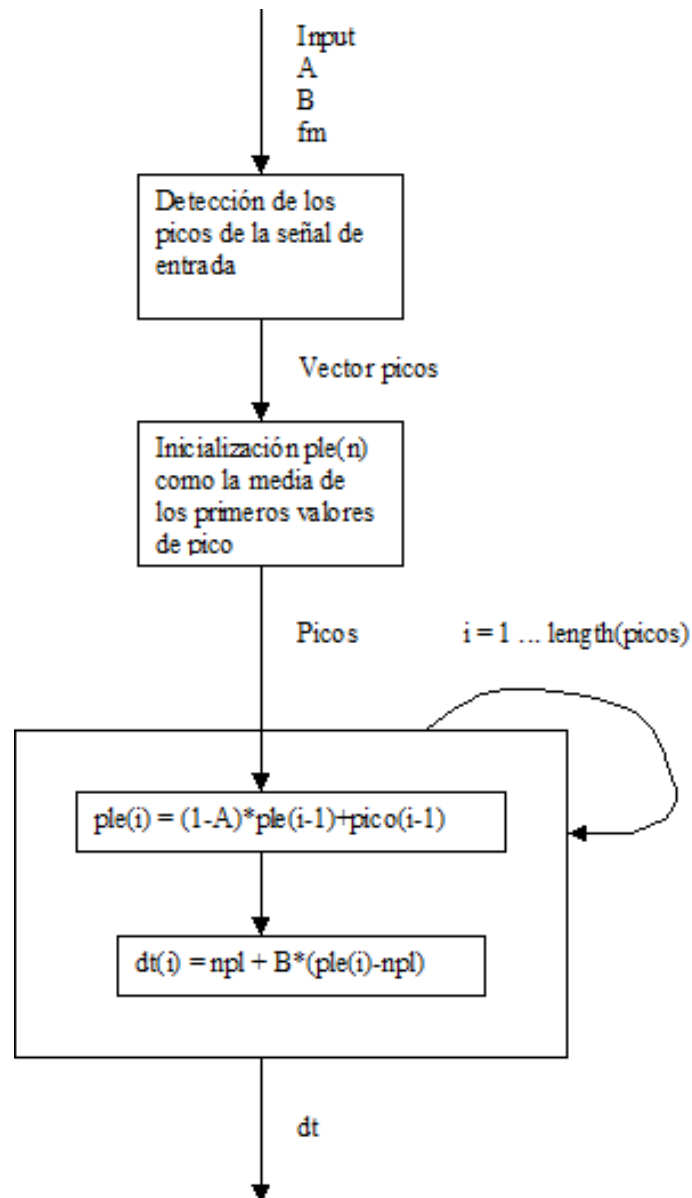


Fig. 34 Flujograma de la función “estimaciónumbral”.

En primer lugar, se obtiene una estimación del nivel de pico en cada pico que dependerá, asimismo, del valor de los picos anteriores. Se trata de una aproximación recursiva que sigue la siguiente regla:

$$\text{Estimación de nivel de pico}(n) = (1-A) \cdot \text{Estimación de nivel de pico}(n-1) + A \cdot \text{Pico}(n)$$

donde A es la sensibilidad hacia nuevos picos.

A partir de esta estimación, se calcula el vector de umbrales de decisión mediante la siguiente expresión:

$$\text{Umbral de Decisión } (n) = B \cdot \text{Estimación de nivel de pico}(n)$$

donde B es el coeficiente de sensibilidad que deberá elegir el usuario. A y B se han determinado empíricamente y se les ha asignado 0.5.

-sobrepasaumbral: esta función realiza un barrido de todos los umbrales de detección que previamente se han establecido con la función “estimacionumbral” comprobando que picos de la señal que anteriormente se había procesado superan el umbral de detección¹³. El diagrama de bloques es el que sigue:

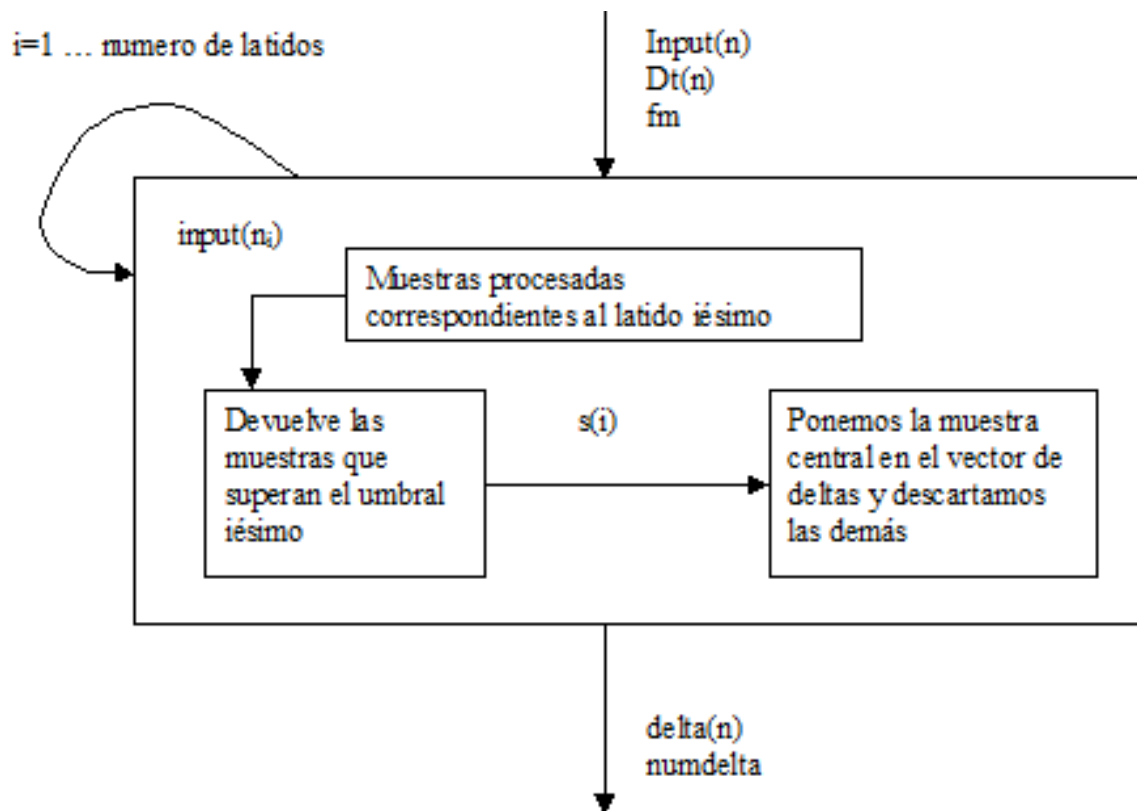
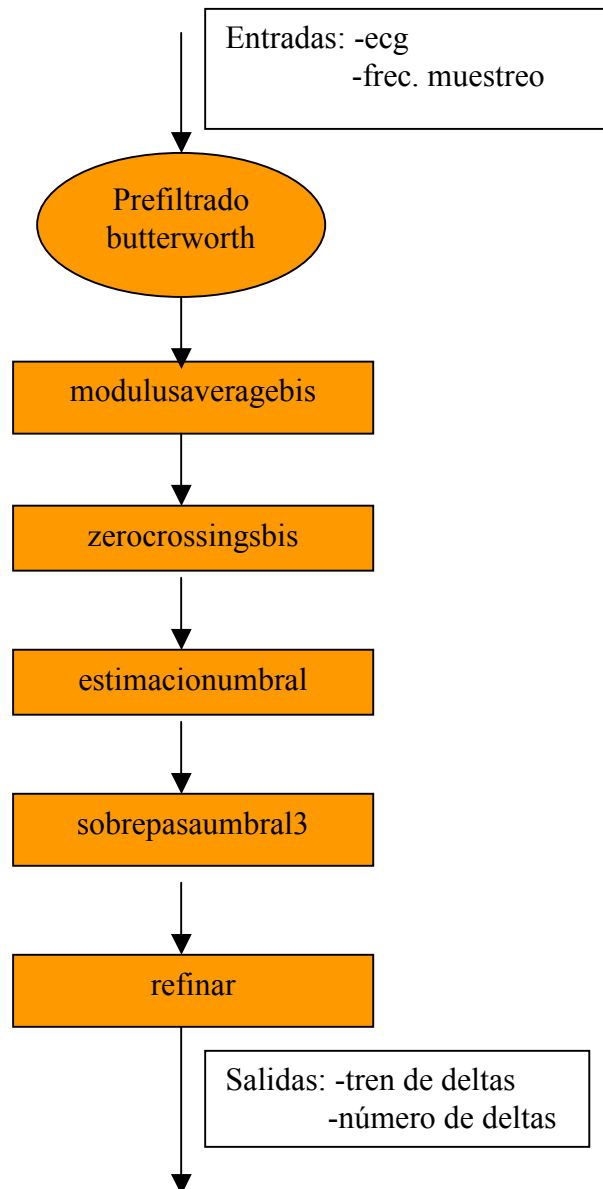


Fig. 35 Flujograma de la función “sobrepasaumbral”.

3.4 Detector de ECGs mediante cruces por cero

Nuestra realización se basa en la propuesta que aparece en el artículo de Köhler, Hennig y Orglmeister (2002)¹⁰.

Dicho algoritmo detecta las ondas QRS mediante la cuenta de los cruces por cero que sufrirá la señal una vez se le haya sumado una señal de alta frecuencia. Sus bloques son los siguientes:



-prefiltrado: elimina el ruido de fuera de la banda de interés por medio de un filtro Butterworth de orden 2 y frecuencias de corte 10 y 30 Hz.

-modulusaveragebis: genera la señal de alta frecuencia que se sumará al ECG.

$$y_2(n) = y_1(n) + b(n) \text{ con } b(n) = k(n) \cdot (-1)^n$$

$b(n)$, señal de alta frecuencia (ver figuras 37 y 38), donde $k(n)$ es la media del módulo del ECG filtrado. El funcionamiento de “modulusaveragebis” es el que siguiente:

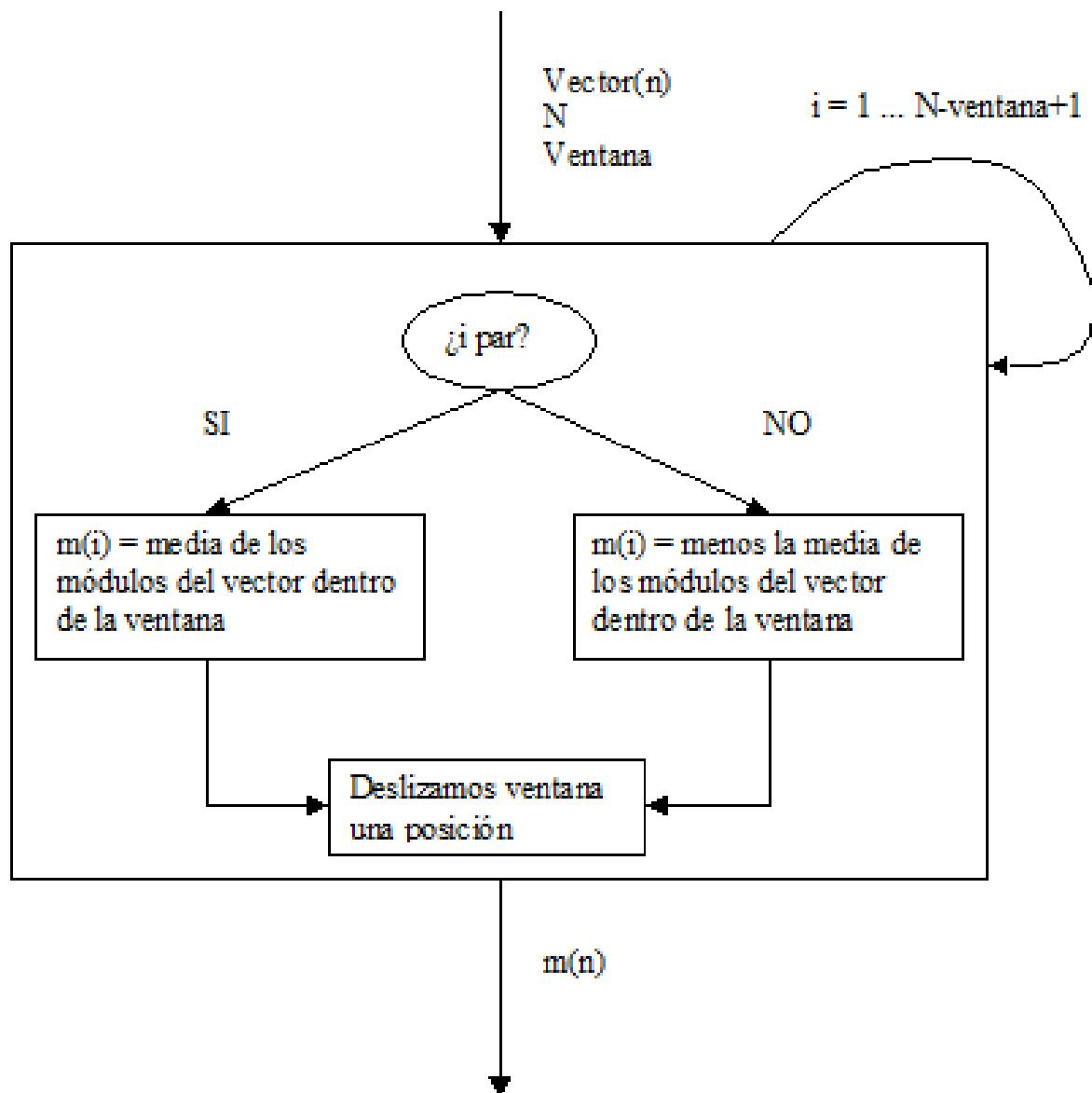


Fig. 36 Flujograma de la función “modulusaveragebis”.

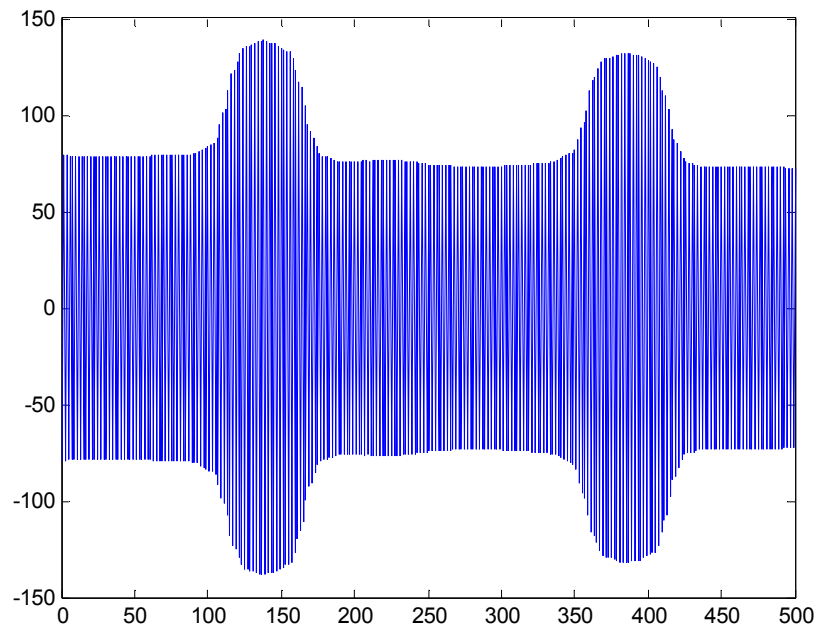


Fig. 37 Señal $b(n)$. Cada punto es la media de los módulos del ECG dentro de una ventana

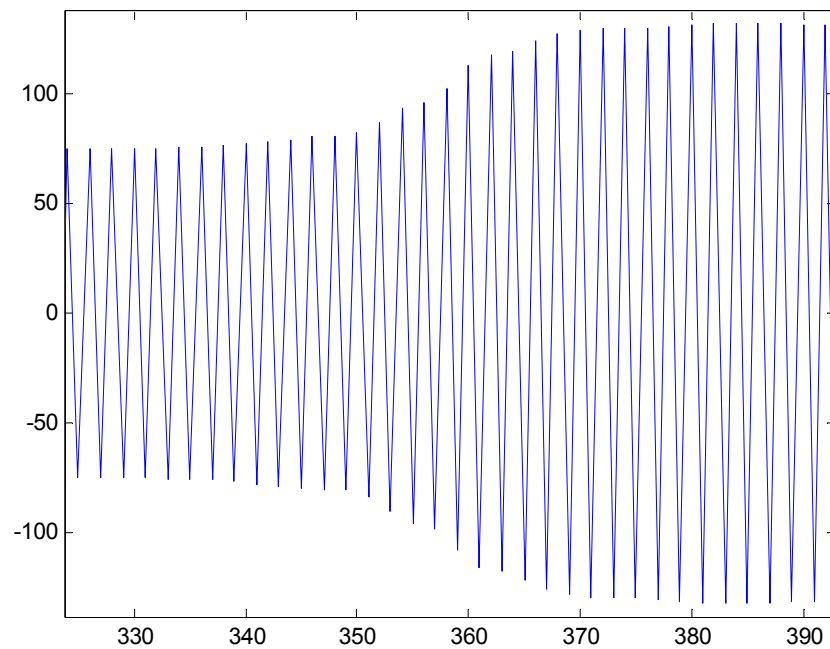


Fig. 38 Señal $b(n)$ de alta frecuencia.

En las figuras 39 y 40 podemos observar como afecta la adición de la señal de alta frecuencia $b(n)$ sobre el ECG.

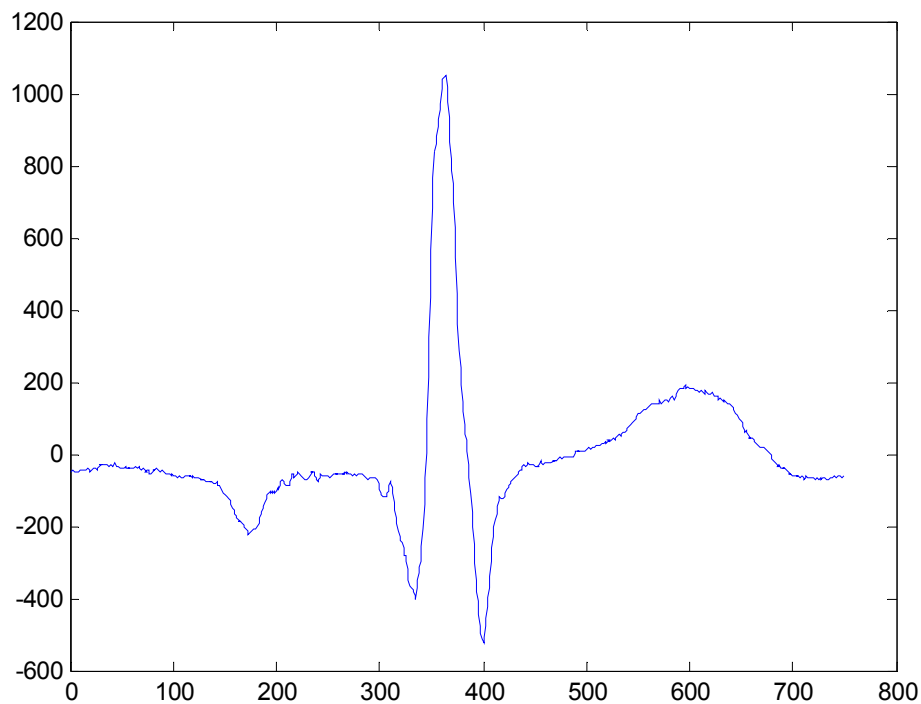


Fig. 39 ECG sin señal de alta frecuencia.

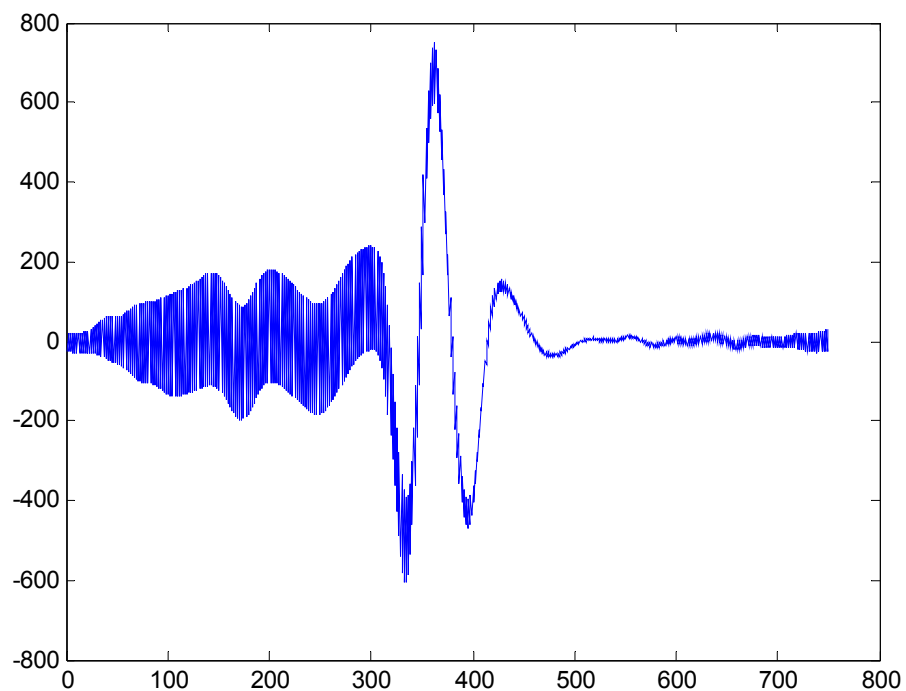


Fig. 40 Señal resultante de la suma del ECG y $b(n)$.

-zerocrossingsbis: genera una señal que es el promedio de los cruces por cero de la señal dentro de una ventana que se va deslizando. En los picos QRS es donde habrá mínimo número de cruces por cero. Nuestra ventana es de 50 muestras dada una $f_m=250$ Hz. Su diagrama de bloques es el que sigue:

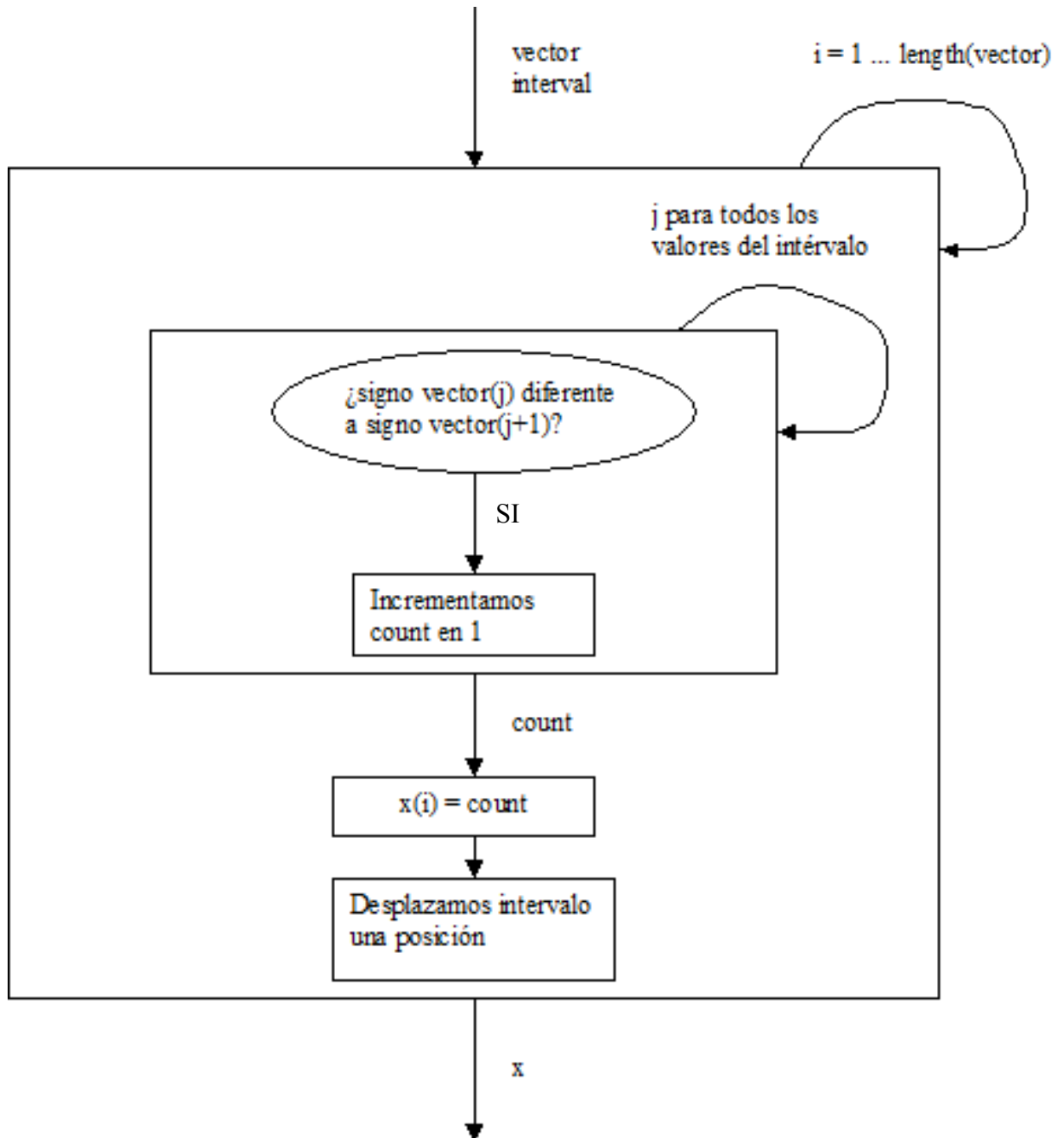


Fig. 41 Flujograma de la función “zerocrossingsbis”.

En la figura 42 se puede apreciar como a partir de la inversa de la señal de promedios de cruces por cero se puede extraer la información de la localización de los picos R.

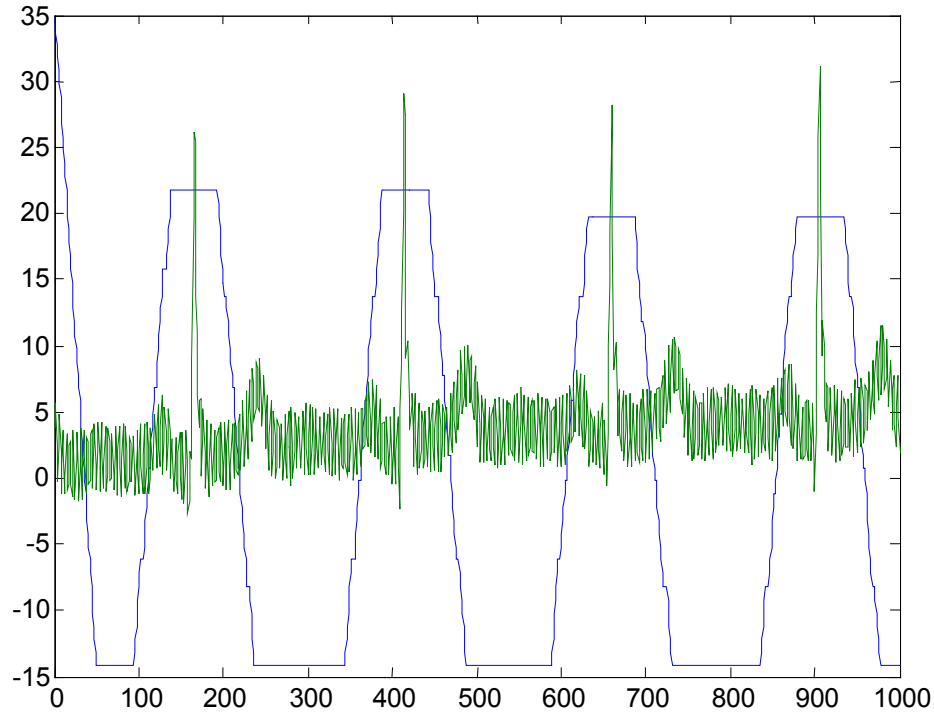


Fig. 42 Señal $z(n)$ en azul (Promediado de los cruces por cero de la suma entre ECG y $b(n)$). ECG en verde.

-estimación umbral: se trata de una función que obtiene un vector de umbrales estimados a través de un algoritmo adaptativo¹³. En primer lugar, se obtiene una estimación del nivel de pico en cada pico que dependerá, asimismo, del valor de los picos anteriores. Se trata de una aproximación recursiva que sigue la siguiente regla:

$$\text{Estimación de nivel de pico}(n) = (1-A) \cdot \text{Estimación de nivel de pico}(n-1) + A \cdot \text{Pico}(n)$$

donde A es la sensibilidad hacia nuevos picos.

A partir de esta estimación, se calcula el vector de umbrales de decisión mediante la siguiente expresión:

$$\text{Umbral de Decisión}(n) = B \cdot \text{Estimación de nivel de pico}(n)$$

donde B es el coeficiente de sensibilidad que deberá elegir el usuario. A y B se han determinado empíricamente y se les ha asignado 0.5.

A continuación se representa su diagrama de bloques:

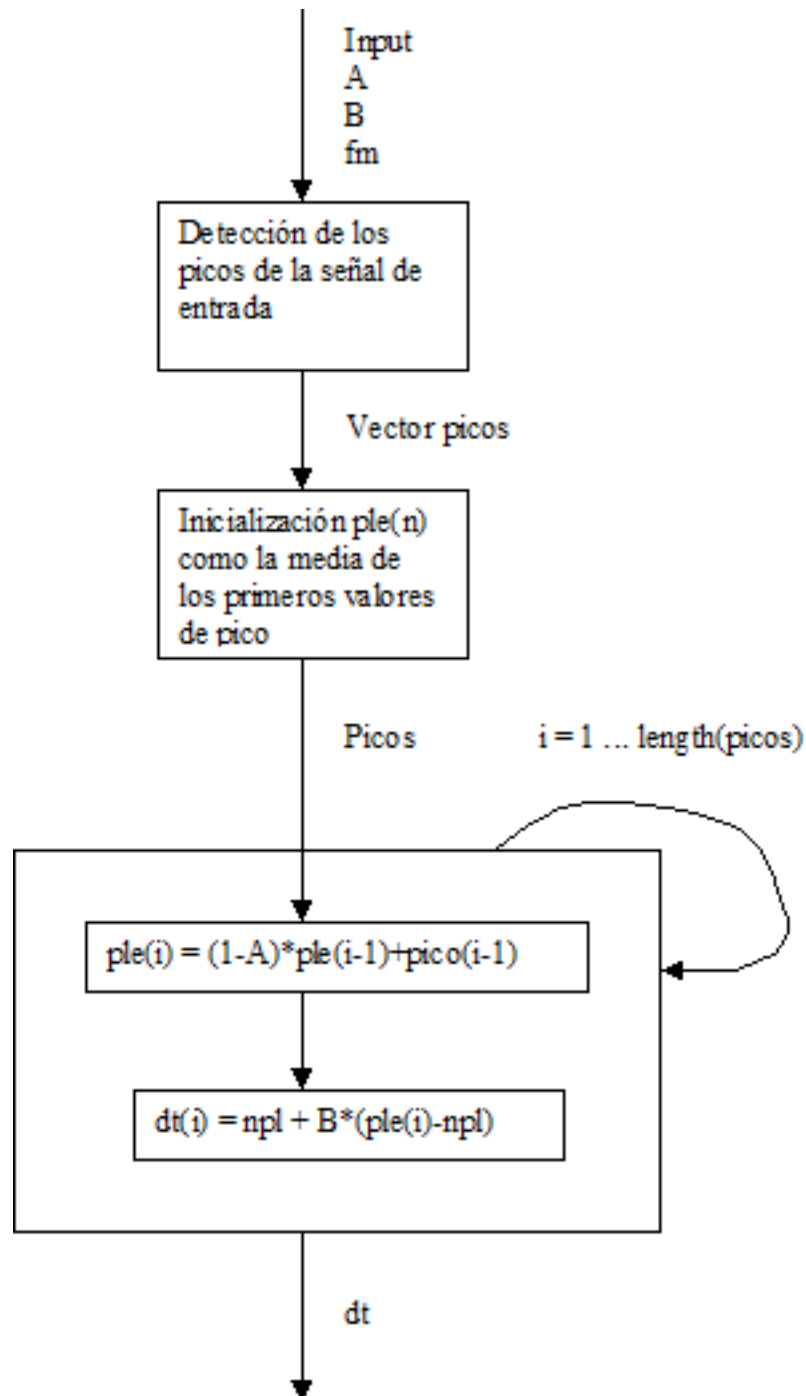


Fig. 43 Flujograma de la función “estimacionumbra”.

-sobrepasaumbral: esta función realiza un barrido de todos los umbrales de detección que previamente se han establecido con la función “estimacionumbral” comprobando que picos de la señal que anteriormente se había procesado superan el umbral de detección. El diagrama de bloques es el que sigue:

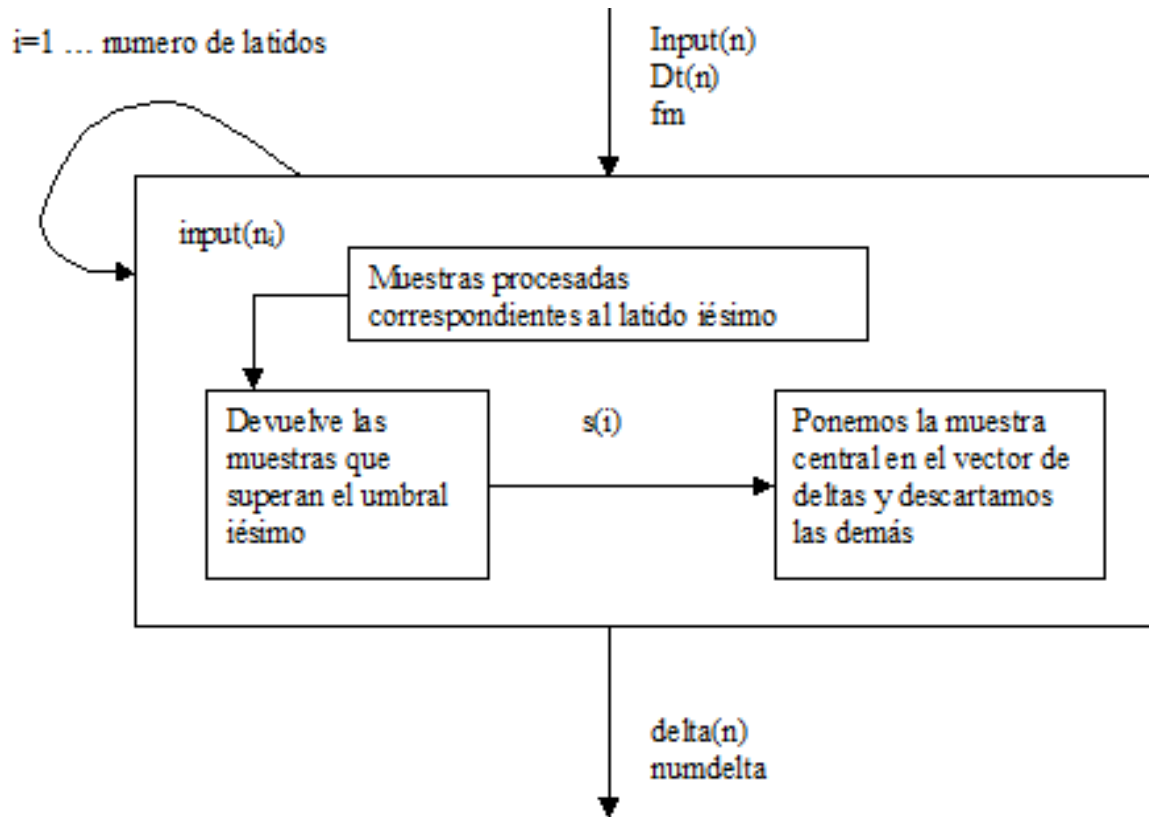


Fig. 44 Flujograma de la función “sobrepasaumbral”.

-refinar: a partir del tren de deltas obtenido, se busca dentro del ECG el máximo allá donde se encuentren unos. Así pues, a estos nuevos máximos se les asignará en un nuevo tren de deltas (trenSalida(n)).

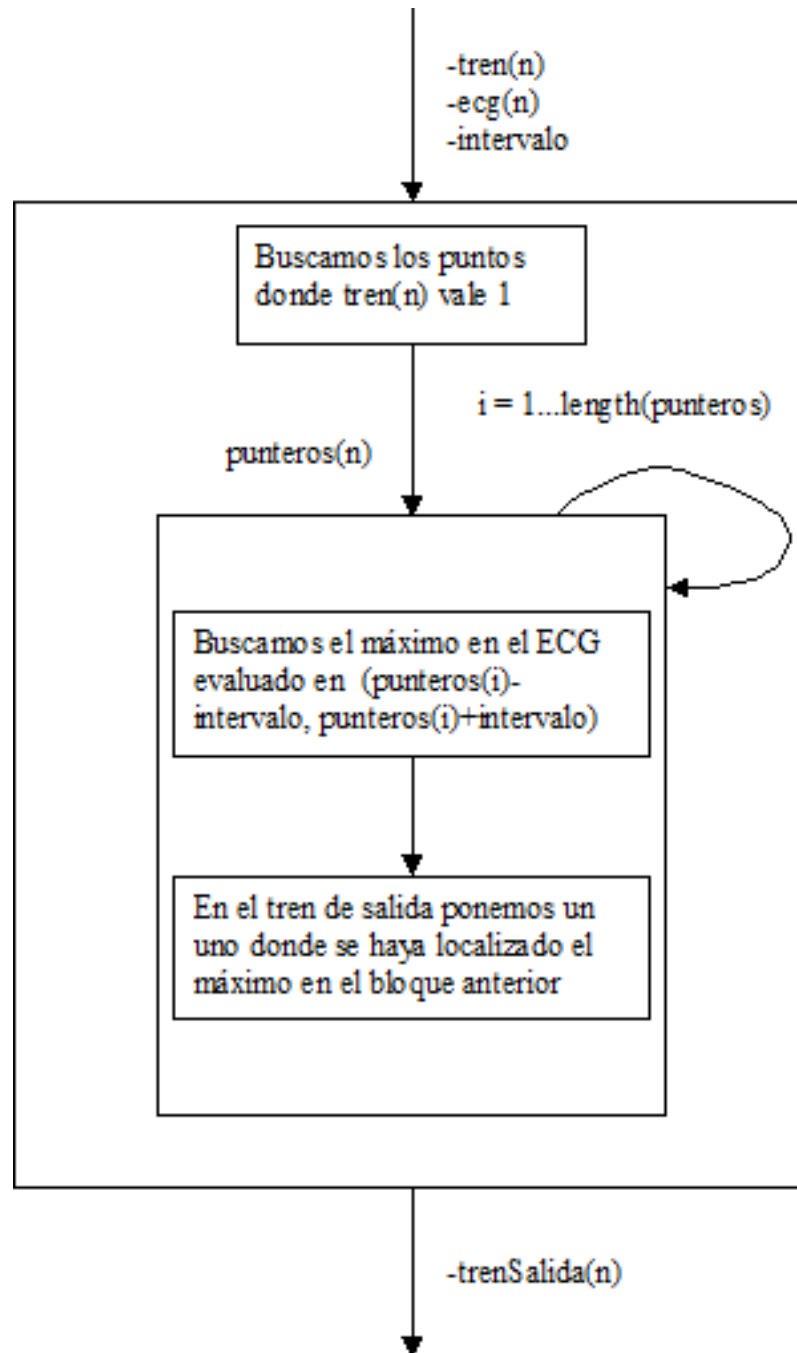


Fig. 45 Flujograma de la función “refinar”.

4. Evaluación y comparación de los diferentes detectores

La utilización de los algoritmos de detección de QRS dentro de las aplicaciones médicas requiere una evaluación previa de su rendimiento. Los dos principales parámetros que se emplean para dicha evaluación son la sensibilidad (*Se*) y la predictibilidad positiva (+*P*). Su definición es la siguiente:

$$Se = \frac{TP}{TP + FN}$$

$$+P = \frac{TP}{TP + FP}$$

donde *TP* son las detecciones positivas, *FN* el número de falsos negativos y *FP* el número de falsos positivos.

Así pues, entendemos por detección positiva aquella delta que representa un pico R que realmente existe en el ECG. Por el contrario, falso negativo correspondería a la no detección de un pico. Por último, un falso positivo consiste en la detección de un pico R allí donde no existe ninguno.

En este capítulo de nuestro estudio evaluaremos los distintos algoritmos elaborados anteriormente (incluiremos también los algoritmos ya implementados en otro proyecto y que aprovecharemos) en base a estos dos parámetros con el objeto de comprobar su correcto funcionamiento.

Una vez realizada la detección de los complejos QRS, hemos procedido a la elaboración de la serie RR. Para ello, hemos empleado la función “diferenciaentrelatidos” que montará sobre un vector de salida una secuencia donde cada punto es la diferencia de muestras existente entre la delta actual y la anterior. La figura 46 nos presenta una serie RR:

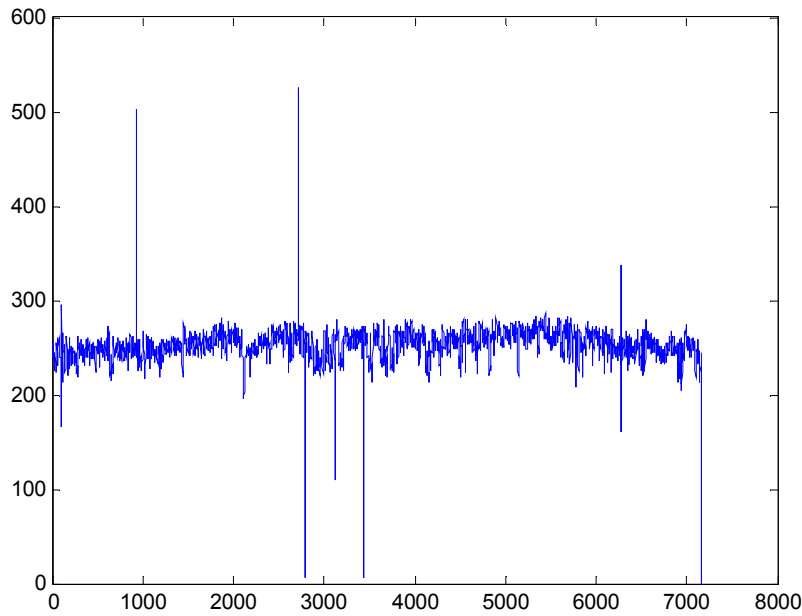


Fig. 46 Serie RR obtenida por medio de un detector wavelet

A partir de la serie RR se pueden localizar con mayor facilidad los falsos negativos y los falsos positivos (figura 47). Los primeros, suelen provocar un pico que despunta hacia arriba de la media del RR debido a que al no poner una delta en un sitio donde debería haber una, lo que se provoca es un drástico incremento del número de muestras de distancia entre dos deltas detectadas (figura 48).

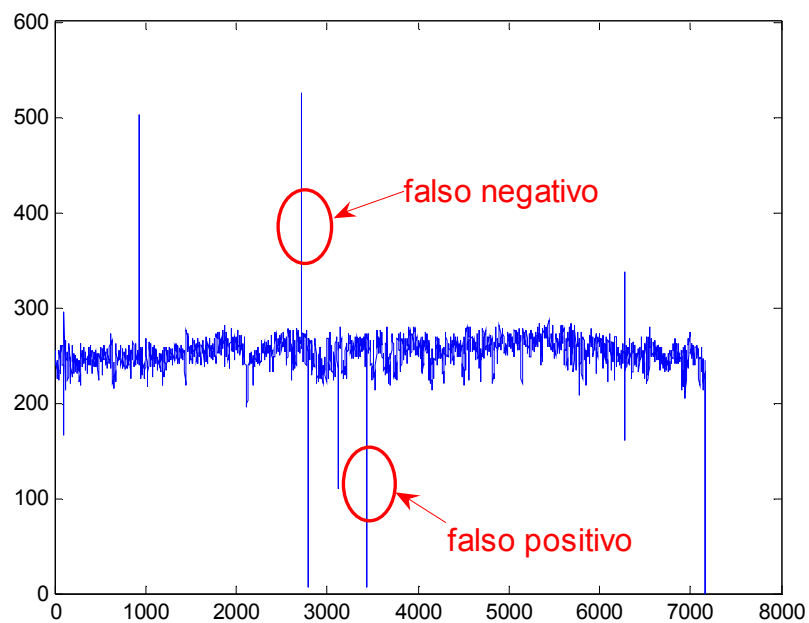


Fig. 47 Localización de falsos positivos y falsos negativos en la serie RR.

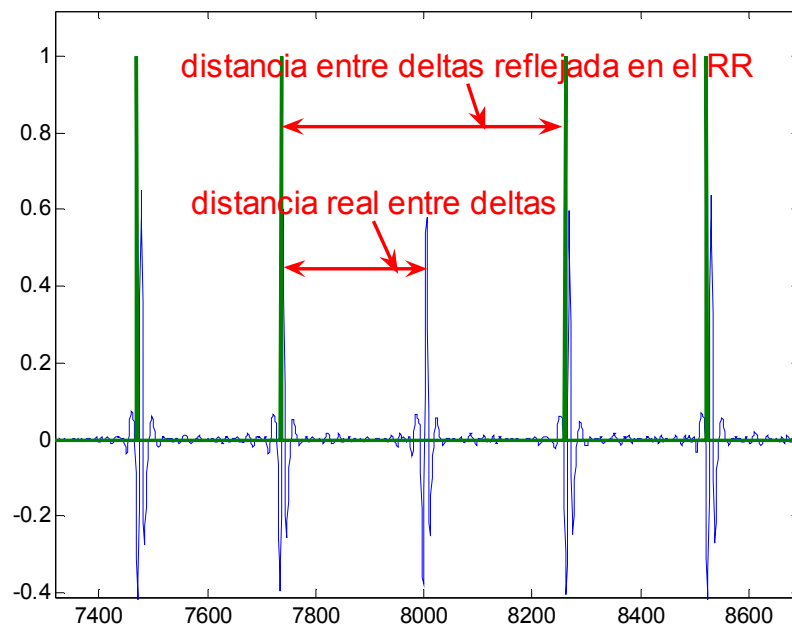


Fig. 48 El falso negativo provoca un aumento notable de la distancia entre deltas con respecto a la distancia real.

En cambio, los falsos positivos provocan un pico hacia abajo ya que intercalando una delta que no procede entre otras dos bien detectadas provoca un decremento súbito de la distancia entre deltas (figura 49).

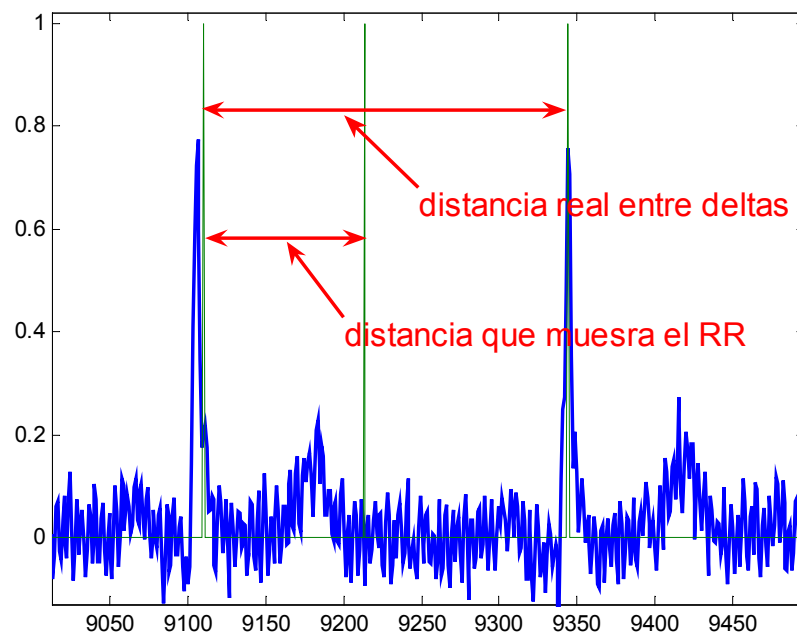


Fig. 49 El falso positivo provoca un decremento notable en la distancia entre deltas con respecto a la distancia real.

La siguiente tabla nos indica los distintos valores de estos parámetros para cada detector en el estudio de 40 casos de ECGs de jóvenes (de entre 21 y 34 años) y de personas de la tercera edad (de entre 68 y 85 años) cuya buena salud ha sido rigurosamente contrastada (se han obtenido del banco de ECGs fantasía¹¹). La forma ideal de detectar los falsos positivos y los falsos negativos sería la de superponer ECG y tren de deltas y verificar delta a delta que la detección haya sido correcta. En nuestro caso, sin embargo, dada la longitud del ECG de partida (1812903 muestras a 250 Hz o, lo que es lo mismo, unas 2 horas de captura de datos) se ha optado por implementar un bloque “performanceDetection”. Dicha función tendrá como entrada la serie RR de la detección y dispondrá a la salida los parámetros P y Se ya calculados. El bloque calcula la media del RR y posteriormente considera falso negativo todo punto que sobrepase 1.3 veces la media y falso positivo todo punto inferior a 0.7 veces la media (valores elegidos por vía experimental). Se ha añadido los resultados obtenidos para los otros 5 detectores que emplearemos más adelante en el proyecto. Como se puede apreciar, los cuatro detectores tienen una sensibilidad y una predictibilidad más que correcta (mayor que el 99 %).

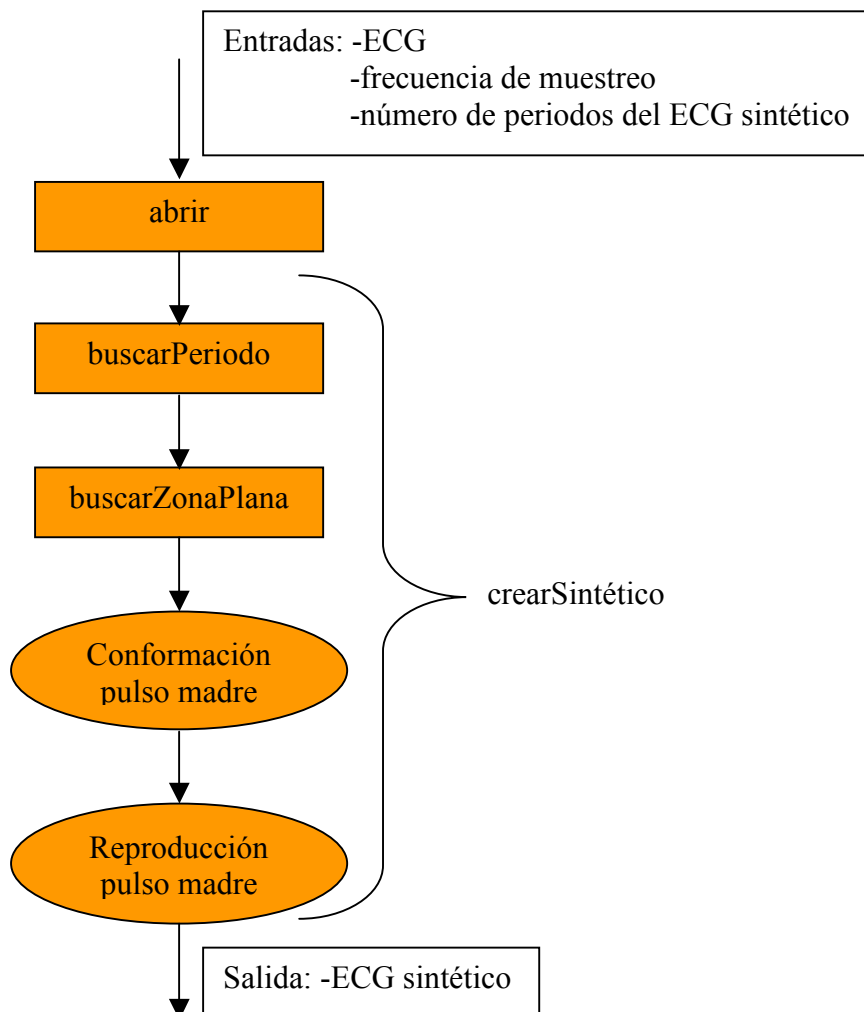
	Wavelet Trans.		Hilbert Trans.		Zero Crossings		Genetic Alg.		H-Tompkins		Ruha		Trahanias		Length Transf.		Energy Transf	
	Se	+P	Se	+P	Se	+P	Se	+P	Se	+P	Se	+P	Se	+P	Se	+P	Se	+P
F1o01	1	0,999	0,999	0,999	0,998	0,999	0,999	0,998	0,999	0,999	0,999	1	0,999	0,999	0,999	0,999	0,999	0,999
F1o02	0,998	0,998	0,995	0,999	0,995	0,999	0,996	0,999	0,996	1	0,996	0,999	0,996	0,999	0,996	0,999	0,996	1
F1o03	0,999	0,998	0,997	1	0,998	0,999	0,997	0,995	0,998	1	0,999	1	0,998	0,998	0,997	1	0,998	1
F1o04	0,999	0,997	0,999	0,999	0,998	0,999	0,993	0,991	0,997	0,999	0,997	0,999	0,998	0,998	0,995	0,999	0,997	0,999
F1o05	1	0,999	0,999	0,999	0,997	0,999	0,997	0,995	0,999	0,999	0,999	0,999	0,999	0,999	0,998	0,999	0,998	0,999
F1o06	1	0,998	0,996	1	0,998	0,999	0,998	0,997	0,998	0,999	0,998	1	0,997	0,999	0,997	0,999	0,998	1
F1o07	0,995	0,996	0,99	0,994	0,998	0,999	0,987	0,994	0,988	0,999	0,988	1	0,987	0,998	0,986	1	0,987	1
F1o08	1	0,998	0,998	0,999	0,996	0,999	0,998	0,997	0,998	0,999	0,999	1	0,998	0,998	0,998	0,999	0,998	0,999
F1o09	1	0,98	0,959	0,999	0,974	0,999	1	0,98	0,971	0,999	0,956	0,999	0,976	0,997	0,958	0,999	0,967	1
F1o10	1	0,999	0,999	1	0,988	0,993	1	0,999	0,999	1	0,999	1	0,998	0,999	0,998	1	0,999	1
F1y01	0,981	0,999	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
F1y02	0,999	0,989	0,994	0,997	0,995	0,997	0,994	0,997	0,994	0,997	0,995	0,996	0,994	0,996	0,994	0,997	0,994	0,997
F1y03	1	0,999	0,999	1	0,999	0,999	0,999	1	0,999	1	0,999	1	0,999	1	0,999	1	0,999	1
F1y04	1	0,989	0,996	1	0,998	0,999	0,998	0,999	0,997	1	0,99	1	0,997	1	0,997	0,999	0,997	1
F1y05	0,999	0,993	0,993	0,997	0,993	0,997	0,995	0,996	0,995	0,997	0,987	0,999	0,995	0,997	0,995	0,997	0,995	0,997
F1y06	0,999	0,993	0,993	0,999	0,999	0,996	0,993	0,994	0,995	0,999	0,982	0,977	0,992	0,997	0,993	0,999	0,994	0,998
F1y07	0,98	0,96	0,948	0,988	0,976	0,99	0,974	0,988	0,97	0,988	0,913	0,992	0,974	0,991	0,97	0,988	0,97	0,989
F1y08	1	0,996	0,999	0,999	0,998	0,998	0,999	0,999	0,999	0,999	0,99	0,999	0,999	0,999	0,999	0,999	0,999	0,999
F1y09	0,999	0,998	0,998	0,999	0,995	0,997	0,999	0,998	0,999	0,999	0,982	0,999	0,996	0,998	0,998	0,999	0,999	0,999
F1y10	0,99	0,999	0,994	0,998	0,996	0,997	0,994	0,998	0,994	0,998	0,993	0,999	0,994	0,998	0,995	0,998	0,994	0,998
F2o01	0,995	0,988	0,984	0,999	0,986	0,999	0,992	0,989	0,984	0,999	0,984	0,999	0,985	0,999	0,984	0,999	0,984	0,999
F2o02	0,998	0,998	0,995	0,999	0,996	0,999	0,998	0,997	0,995	1	0,995	0,999	0,995	0,998	0,995	0,999	0,995	1
F2o03	0,999	0,999	0,998	1	0,998	0,999	0,999	0,999	0,998	1	0,998	1	0,998	0,999	0,998	1	0,998	1
F2o04	1	0,999	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
F2o05	0,998	0,998	0,992	0,998	0,994	0,996	0,986	0,987	0,984	0,997	0,993	0,995	0,995	0,999	0,993	0,997	0,984	0,996
F2o06	1	0,998	0,999	0,999	0,998	0,999	1	0,999	0,999	1	0,999	1	0,998	0,999	0,999	1	0,999	1
F2o07	1	0,999	0,998	1	1	1	1	0,999	0,998	1	0,998	1	0,999	0,999	0,998	0,999	0,998	1
F2o08	0,921	0,869	0,859	0,972	0,964	0,918	0,937	0,912	0,863	0,971	0,934	0,915	0,927	0,907	0,861	0,969	0,863	0,971
F2o09	0,999	0,998	0,995	0,997	0,996	0,998	0,993	0,992	0,998	0,999	0,996	0,998	0,994	0,996	0,992	0,995	0,997	0,999
F2o10	0,998	0,998	0,994	0,998	0,995	0,999	0,996	0,996	0,995	0,998	0,995	0,999	0,997	0,999	0,995	0,998	0,995	0,998
F2y01	0,999	0,999	0,998	1	0,998	1	0,998	0,999	0,998	1	0,995	0,999	0,998	1	0,998	1	0,998	1
F2y02	0,995	0,999	0,995	0,999	0,997	0,999	0,994	0,997	0,996	0,999	0,991	1	0,996	0,999	0,995	0,999	0,996	0,999
F2y03	1	0,999	0,999	1	0,999	1	0,998	0,998	1	1	1	1	0,999	0,999	0,999	1	1	1
F2y04	0,999	0,999	0,998	0,999	0,998	0,998	0,992	0,992	0,999	0,999	0,998	0,999	0,997	0,998	0,997	0,999	0,999	1
F2y05	0,999	0,997	0,998	0,999	0,998	0,999	0,999	0,999	0,998	0,999	0,999	1	0,998	0,999	0,998	1	0,998	0,999
F2y06	1	0,999	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
F2y07	0,999	0,999	0,999	0,999	1	0,999	0,999	0,999	0,999	0,999	0,996	1	0,999	1	0,999	0,999	0,999	0,999
F2y08	0,997	0,993	0,993	0,999	0,994	0,999	0,993	0,996	0,992	0,999	0,99	1	0,994	0,998	0,993	0,999	0,992	0,999
F2y09	0,952	0,999	0,961	0,958	0,983	0,989	0,912	0,994	0,912	0,995	0,964	0,954	0,971	0,979	0,958	0,957	0,912	0,995
F2y10	0,999	0,998	0,993	0,999	0,997	0,998	0,997	0,998	0,996	0,999	0,985	0,974	0,993	0,998	0,994	0,999	0,996	0,999
Media	0,994	0,993	0,99	0,997	0,994	0,996	0,992	0,994	0,99	0,998	0,989	0,995	0,993	0,996	0,99	0,997	0,989	0,998

5. Creación de ECGs sintéticos

El objetivo de este apartado es el de crear un espacio de experimentación teórico ideal en el que se pueda apreciar los errores en detección debidos exclusivamente a los cambios de morfología del ECG. Para ello se ha creado un ECG sintético consistente en la reproducción periódica de un latido del ECG original. Así, eliminamos la fuente de SDRR debida a la variabilidad del ritmo cardíaco puesto que, idealmente, sería cero.

Los ECG originarios se han extraído de un banco de ECGs *online*¹² que consta de 290 pacientes que han sufrido algunos problemas de corazón. La frecuencia de muestreo es de 1 kHz y se dispone de las tres componentes del vectorcardiograma de forma que realizando rotaciones de éste podrán simularse cambios en la morfología del ECG. La iteración de un mismo latido en las tres componentes debería dar una serie RR constante. Por lo tanto, cualquier variación en esta serie tras realizar rotaciones será, en principio, debida a los cambios en la morfología del QRS.

Los bloques que conforman “main”, que nos posibilitará la generación de ECGs sintéticos, son los siguientes:



-abrir: función que abre los archivos de la base de datos y asigna a x, y, z cada una de las derivaciones del ECG. A continuación presentamos el flujograma:

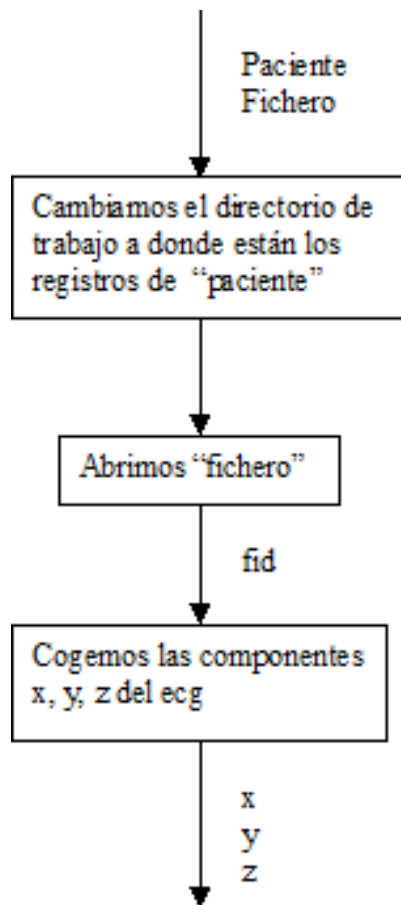


Fig. 50 Flujograma de la función "abrir".

-buscarPeriodo: dados un ECG y una frecuencia de muestreo de entrada hallará el periodo del ECG. El funcionamiento es el siguiente:

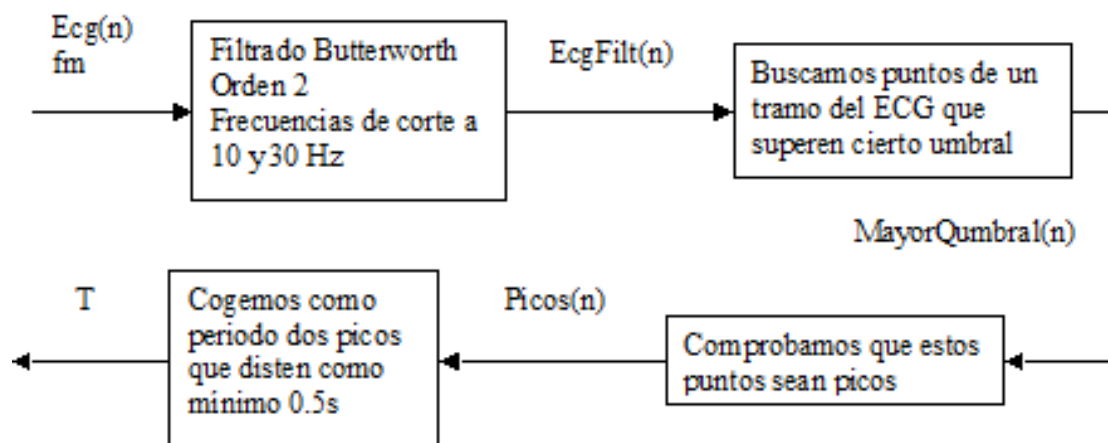


Fig. 51 Flujograma de la función "buscarPeriodo".

-buscarZonaPlana: para evitar discontinuidades en el ECG sintético (que podrían generar falsos positivos) debidos a la inserción de un nuevo periodo (figura 53) trataremos de hallar una zona lo más plana posible desde donde empezar a coger el pulso madre (que será el que reproduciremos). Su diagrama de bloques es el que sigue:

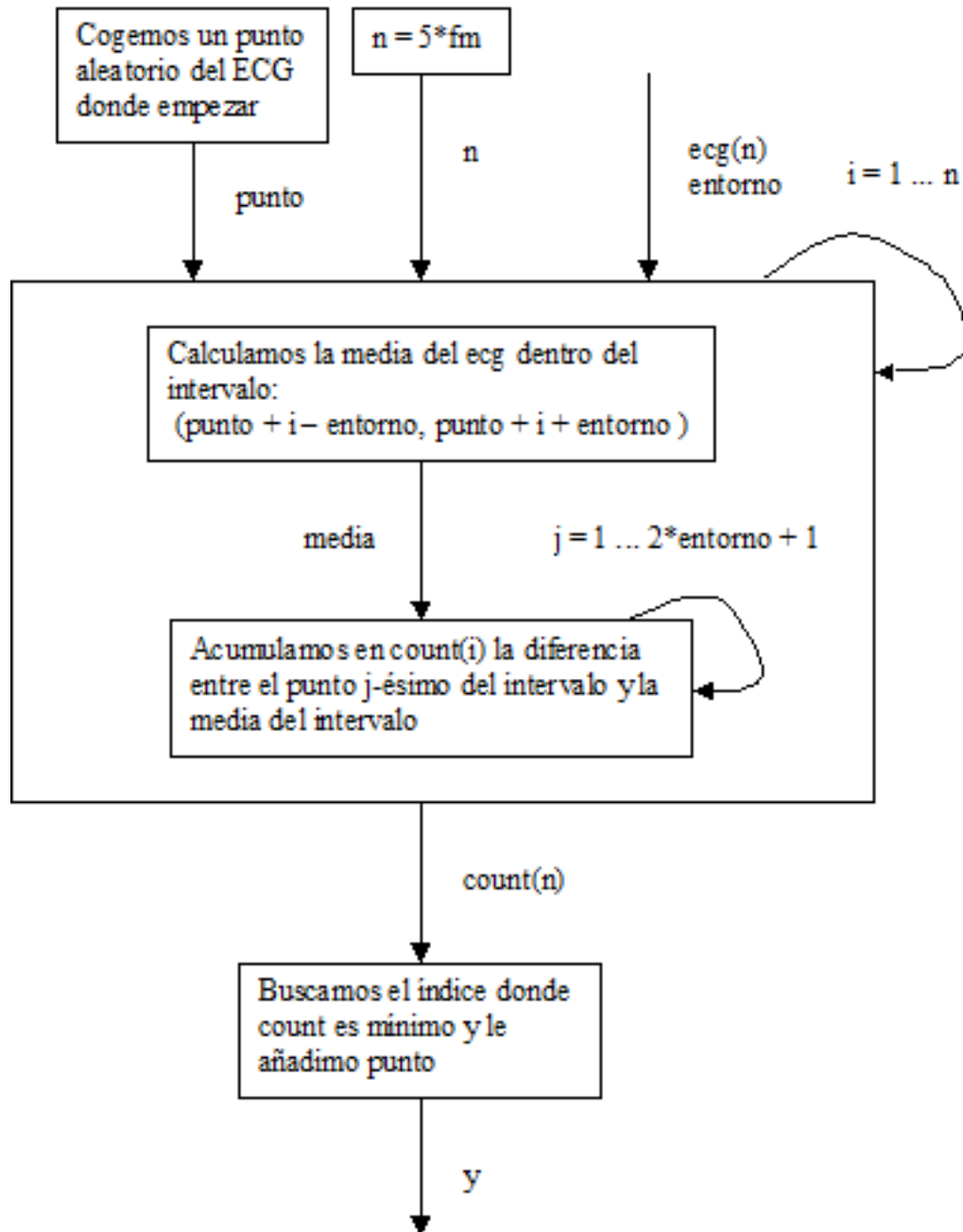


Fig. 52 Flujograma de la función “buscarZonaPlana”.

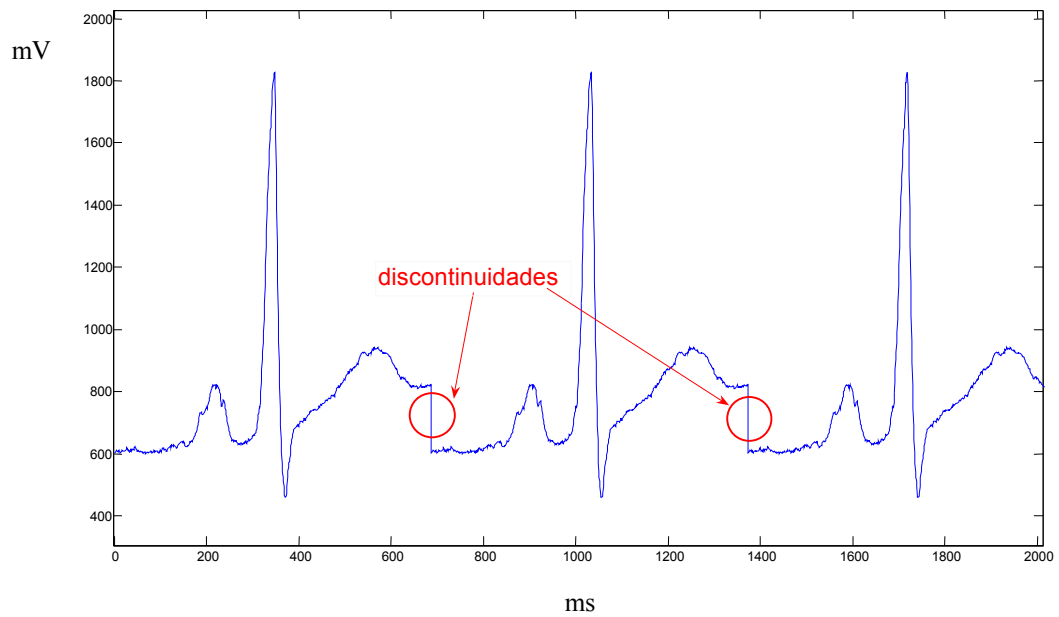


Fig. 53 Discontinuidades en el ECG sintético debidas a la elección del inicio del pulso madre en una zona poco plana.

El resultado de dicha función será el punto del ECG desde donde empezaremos a coger el pulso madre (figura 54).

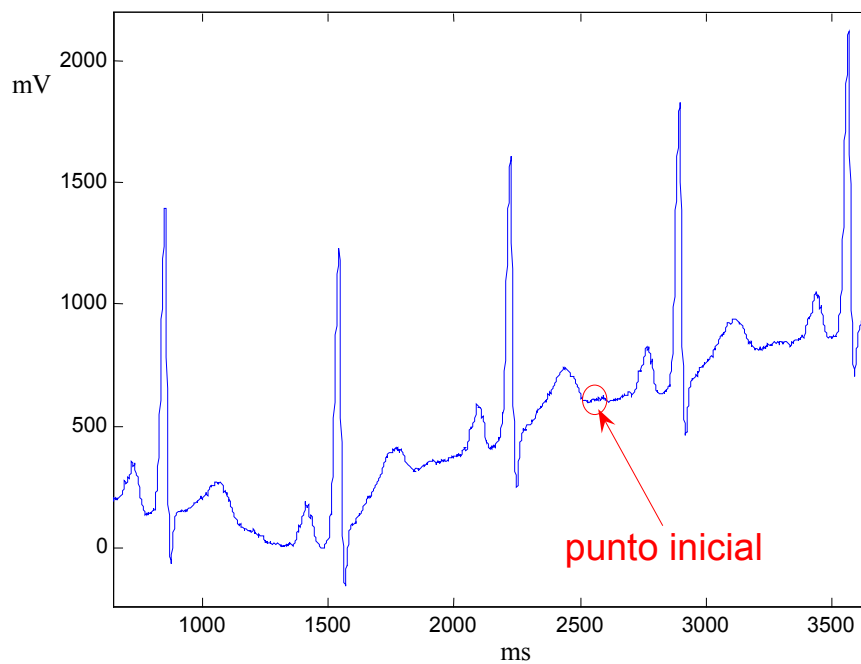


Fig. 54 ECG original y punto inicial para el pulso madre

Una vez hallada una zona más o menos plana (es decir, sin derivas de línea base) y dado el periodo del ECG, cogeremos un pulso madre de la señal (un periodo) para reproducirlo. A dicho pulso madre se le restará la recta que une los dos extremos del intervalo (figura 55) ya que la tendencia creciente o decreciente de la señal también puede producir discontinuidades similares a la descrita anteriormente.

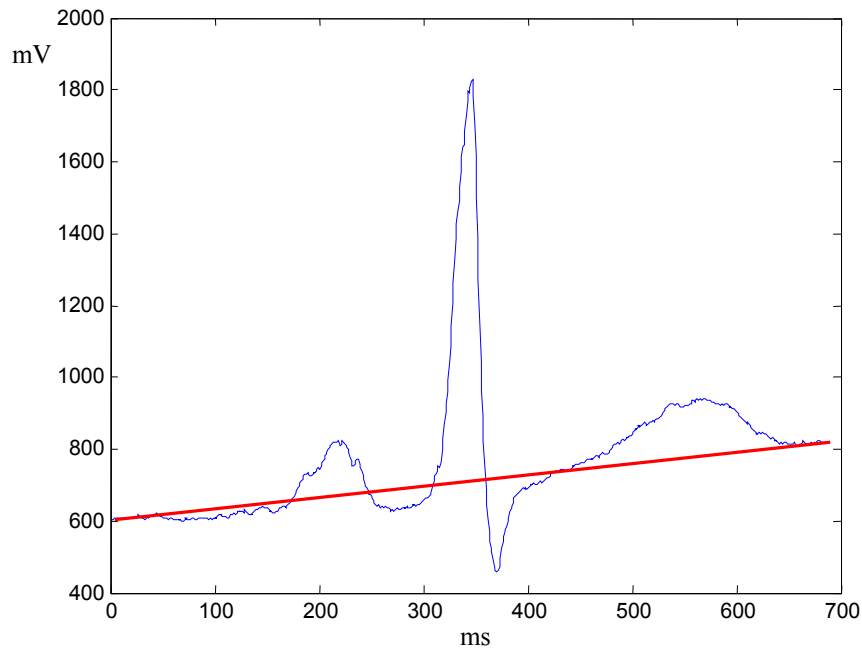


Fig. 55 Pulso madre con tendencia creciente de la señal QRS.

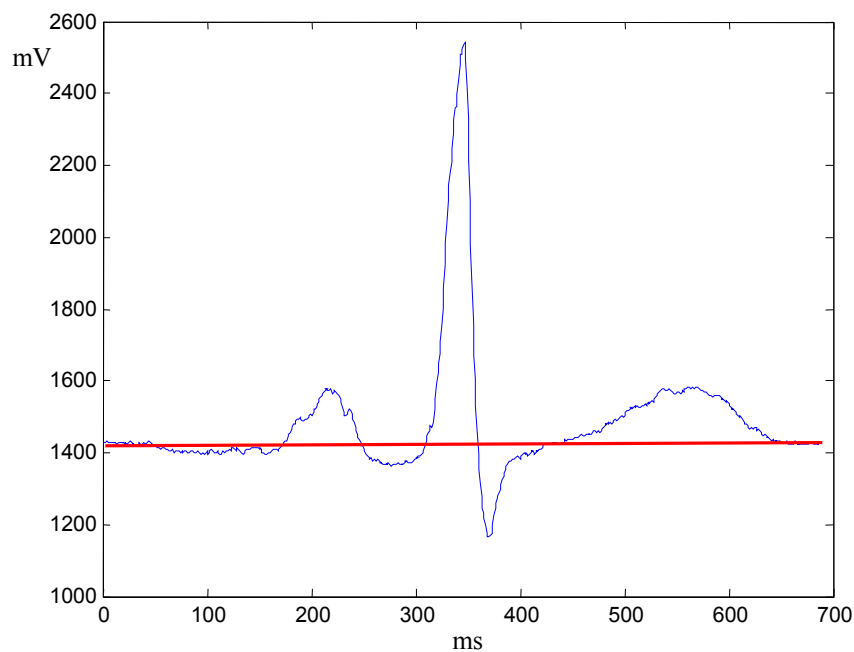


Fig. 56 Pulso madre una vez restada la recta.

Por último, sólo nos queda repetir el pulso madre el número de veces que deseemos y obtendremos el ECG sintético con el que trabajaremos más adelante (figura 57).

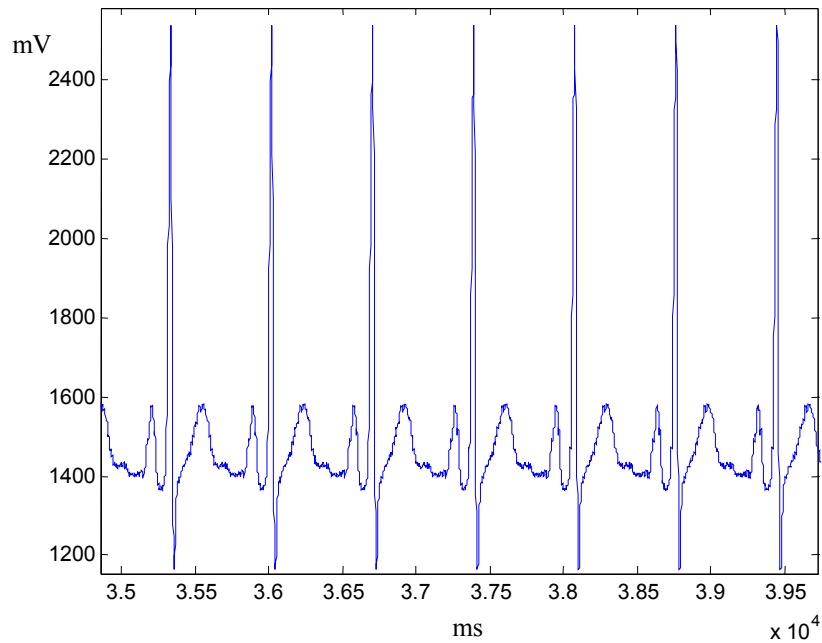


Fig. 57 ECG sintético.

Los ECGs sintéticos creados en el estudio serán reproducciones de 500 latidos a partir del original.

6. Rotación de los electrocardiogramas sintéticos

Una vez obtenido el ECG sintético libre de influencias como la media de base, efectos que alteran la morfología del ECG como podría ser la respiración, etc. modificaremos a nuestro libre albedrío la morfología del ECG mediante la rotación de las diferentes derivaciones. En consecuencia, podremos controlar el grado de variación de las rotaciones con el fin de probar la sensibilidad de los detectores hacia dichos cambios en la morfología del ECG.

Para ello, hemos empleado una función ya implementada: *rotecg*. Se trata de un algoritmo que coge por entrada las tres derivaciones del ECG y la forma de variación propuesta para dicha derivación (expresada en grados). Con todo, a la salida obtendremos las derivaciones resultantes de la rotación del vectorcardiograma.

Las fases empleadas han sido senoides de frecuencia 0.2 Hz (que es la frecuencia por defecto de la respiración para un adulto sano) y cuya amplitud vamos variando según el grado de rotación que deseemos. Así, obtendremos un electrocardiograma que modela los cambios de morfología en el ECG debidos exclusivamente a la respiración.

Las expresiones de las fases aplicadas a cada derivación son las siguientes:

$$fase_x(n) [^\circ] = A_x [^\circ] * \sin(2 * \pi * 0.2 * n / fm)$$

$$fase_y(n) [^\circ] = A_y [^\circ] * \sin(2 * \pi * 0.2 * n / fm)$$

$$fase_z(n) [^\circ] = A_z [^\circ] * \sin(2 * \pi * 0.2 * n / fm)$$

donde A es la amplitud de la fase aplicada y que será nuestro parámetro de juego ya que la magnitud del giro (y, consecuentemente, su peso en la alteración de la correcta detección) será proporcional a él.

Pese a que en algunos artículos salía que la respiración provocaba un máximo de 5 grados de rotación de los ejes hemos trabajado también con amplitudes de 10 grados para dar un margen de error mayor. Los giros practicados son los siguientes:

	A_x [°]	A_y [°]	A_z [°]
giro 1	5	0	0
giro 2	0	5	0
giro 3	5	5	0
giro 4	0	0	5
giro 5	5	0	5
giro 6	0	5	5
giro 7	5	5	5
giro 8	10	0	0
giro 9	0	10	0
giro 10	10	10	0
giro 11	0	0	10
giro 12	10	0	10
giro 13	0	10	10
giro 14	10	10	10

En las siguientes figuras (58, 59 y 60), podemos observar un ejemplo de las tres derivaciones del ECG sintético de un paciente sin rotar y, a continuación, hemos adjuntado las figuras una vez rotadas (con distintas combinaciones en las tres derivaciones). Los ejes de abscisas son milisegundos y las ordenadas son milivoltios.

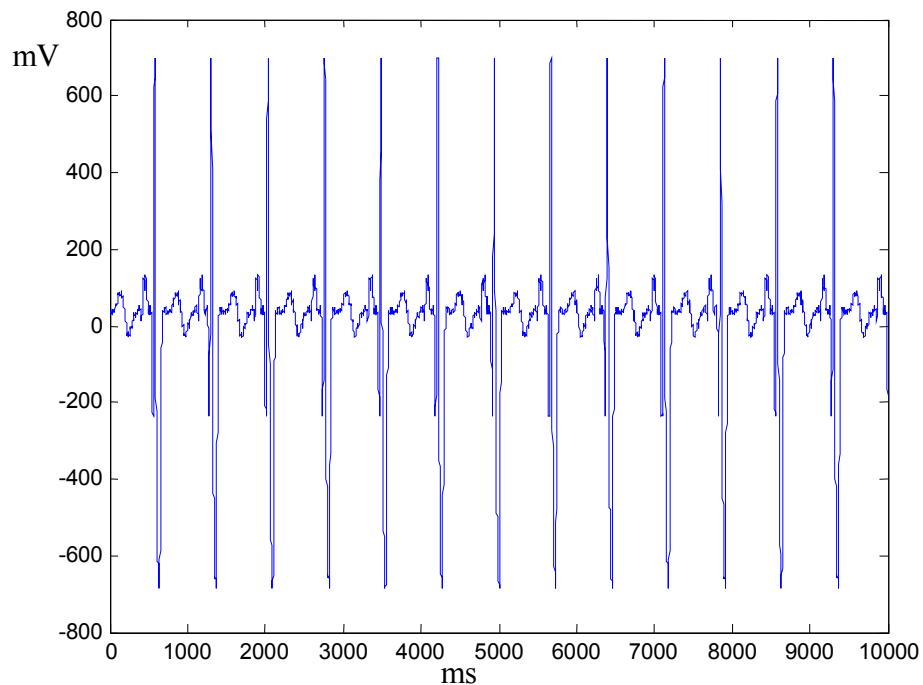


Fig. 58 Derivación x sintética del paciente 1 sin rotar.

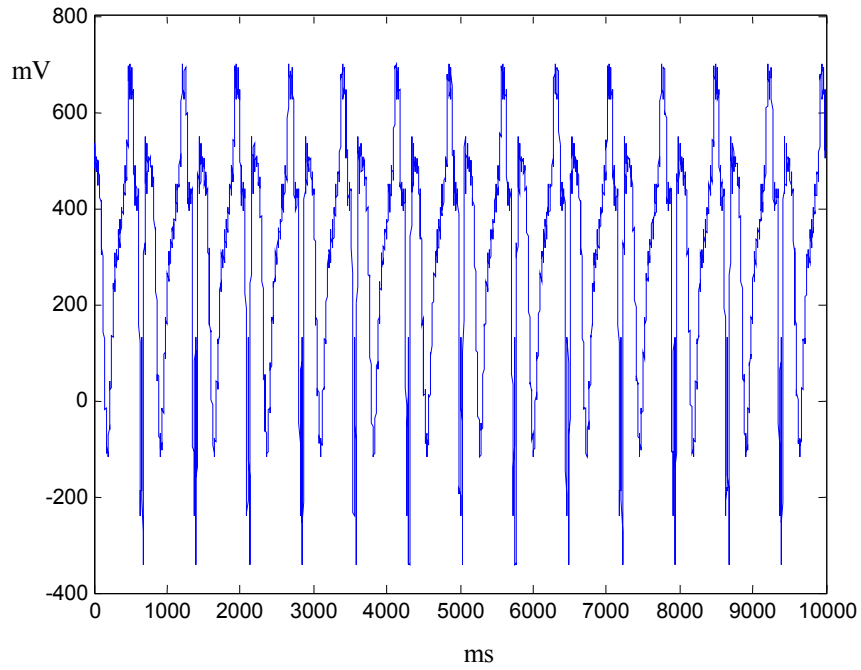


Fig. 59 Derivación y sintética del paciente 1 sin rotar.

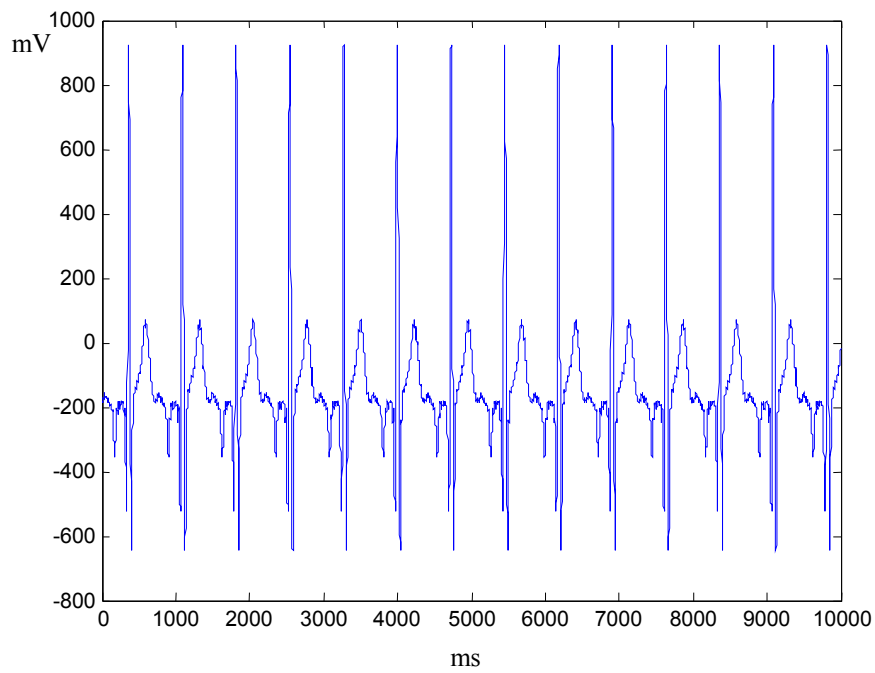
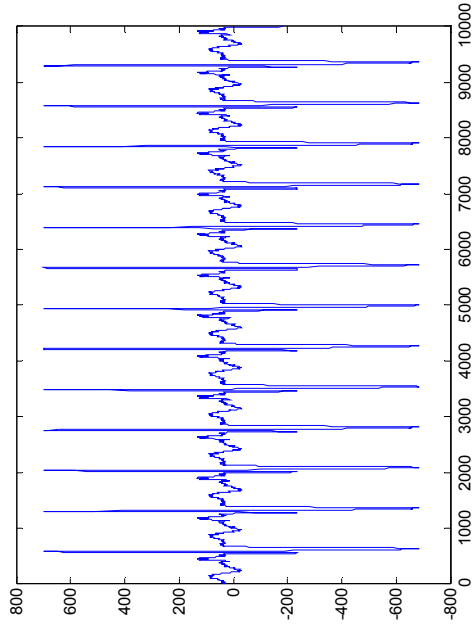
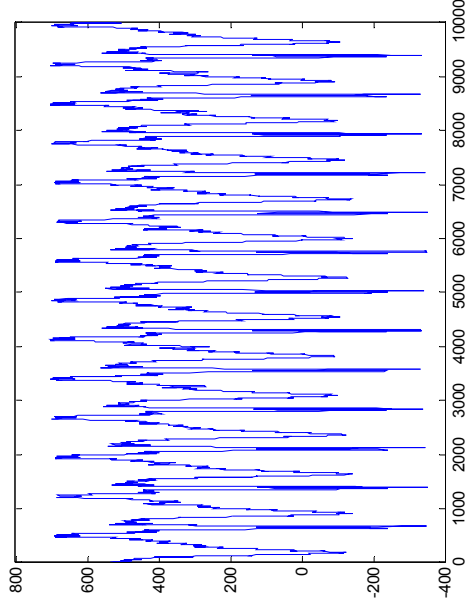


Fig. 60 Derivación z sintética del paciente 1 sin rotar.

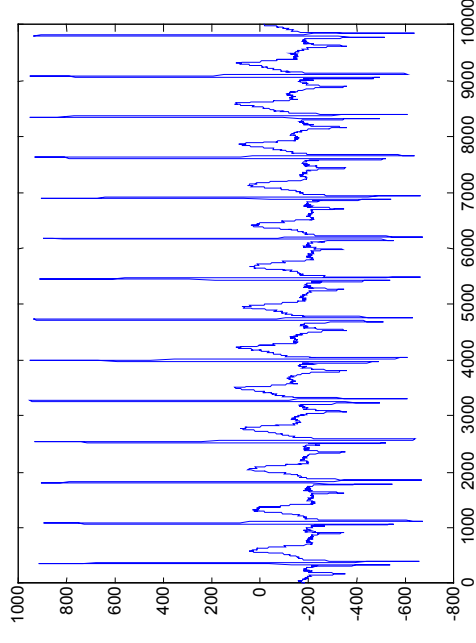
Giro $x=5, v=0, z=0$



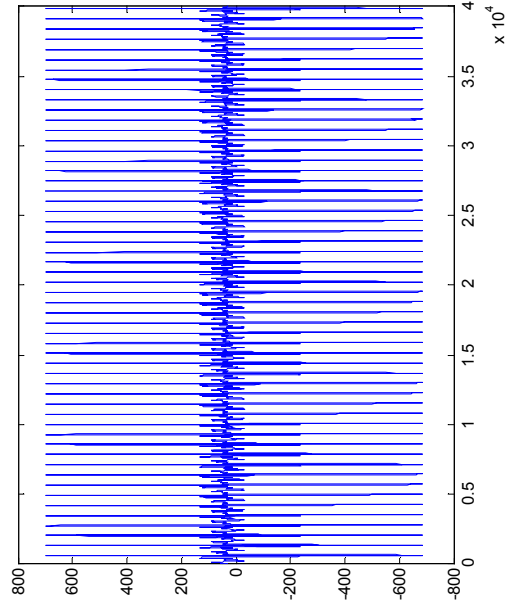
derivación x



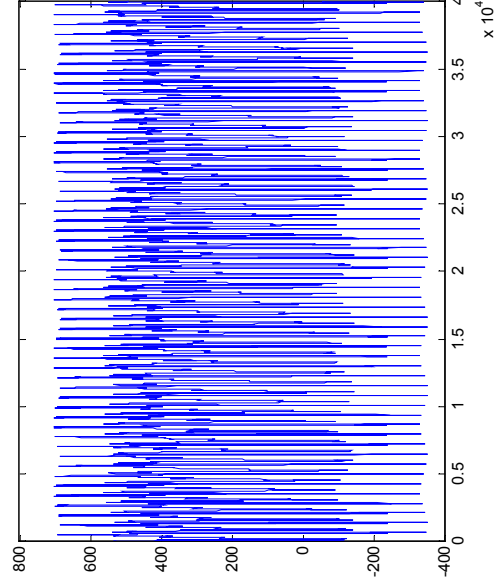
derivación y



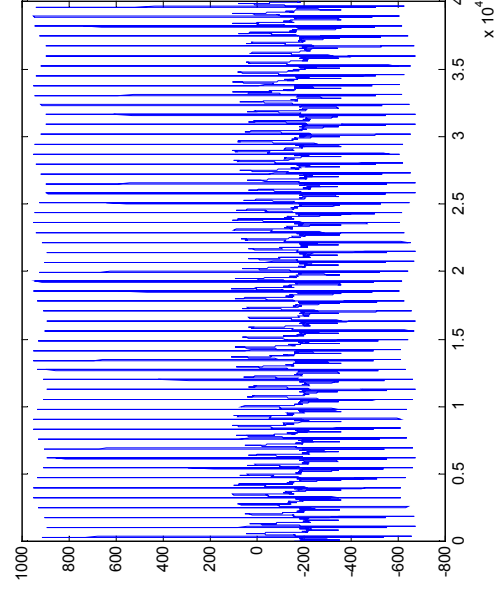
derivación z



derivación x



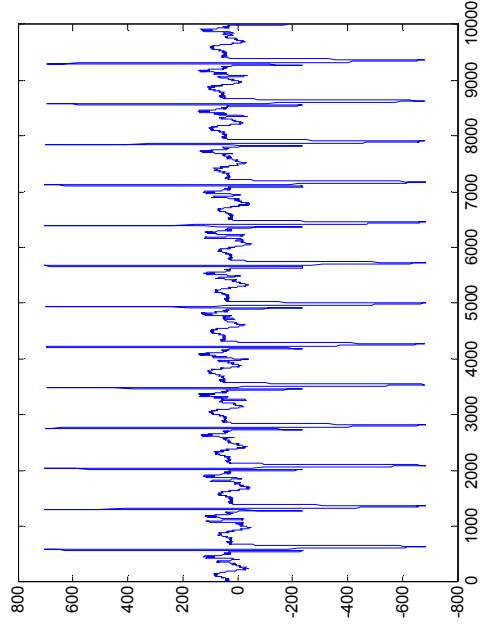
derivación y



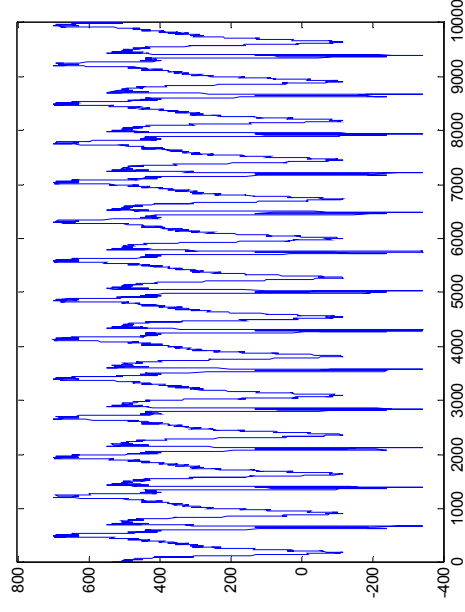
derivación z

z

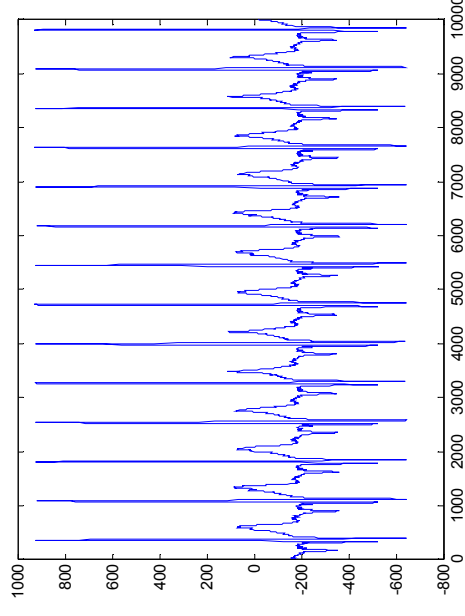
Giro $x=0, y=5, z=0$



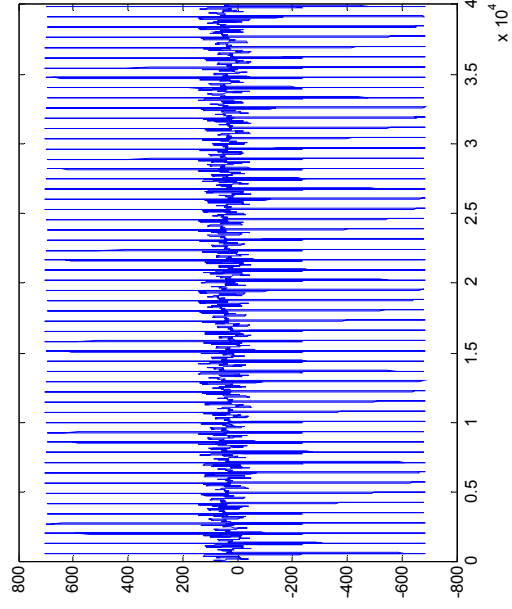
derivación x



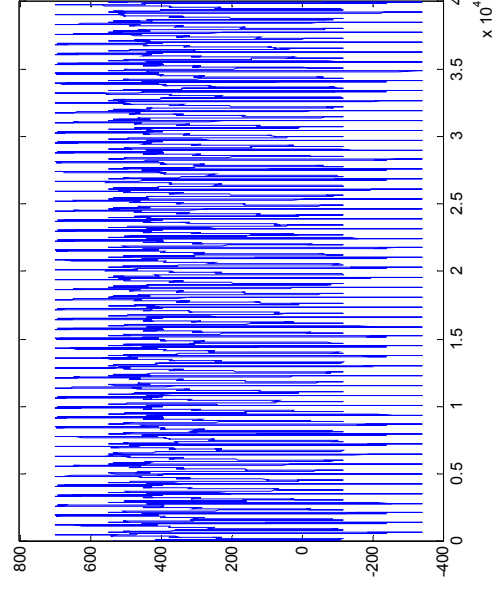
derivación y



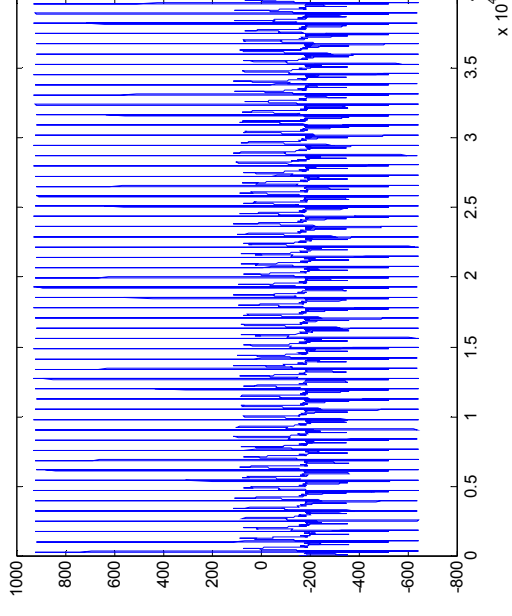
derivación z



derivación x

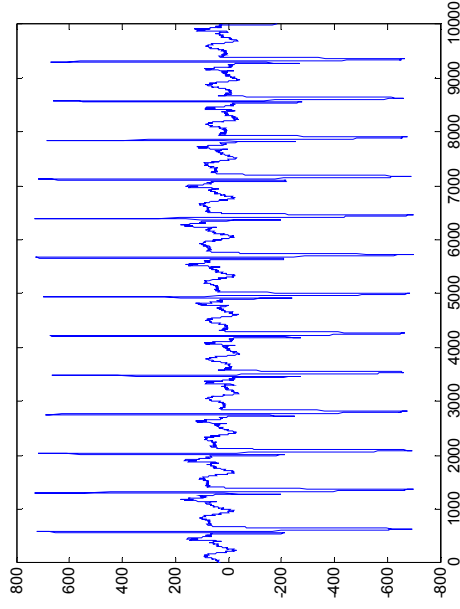


derivación y

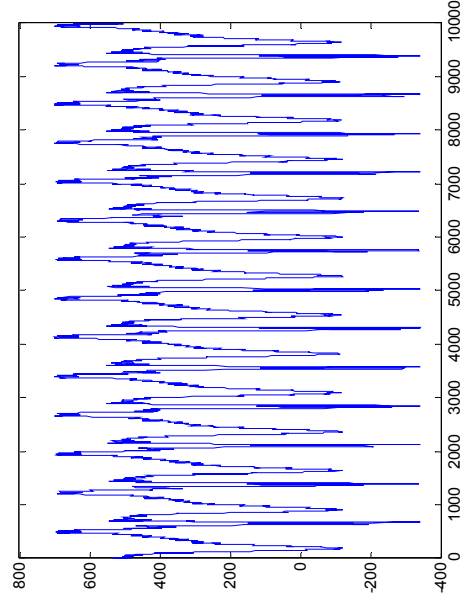


derivación z

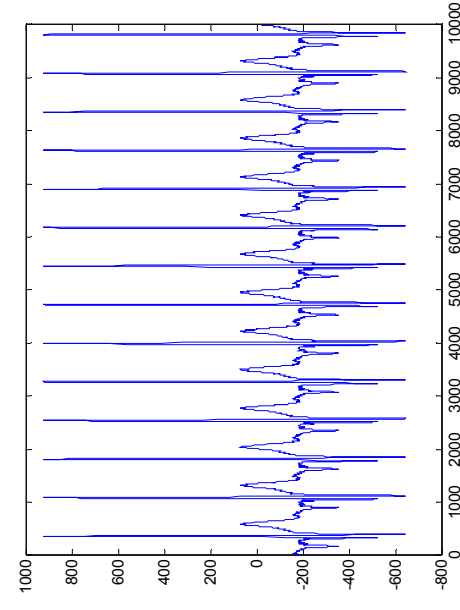
Giro $x=0, y=0, z=5$



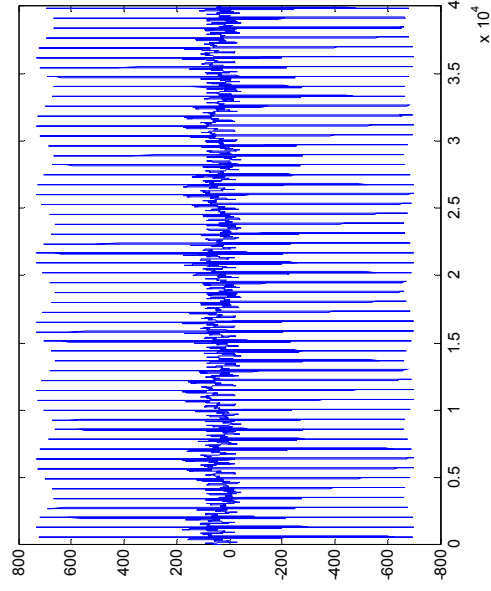
derivación x



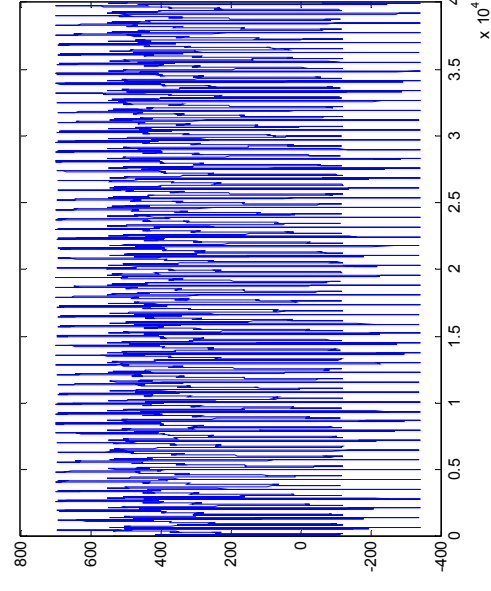
derivación y



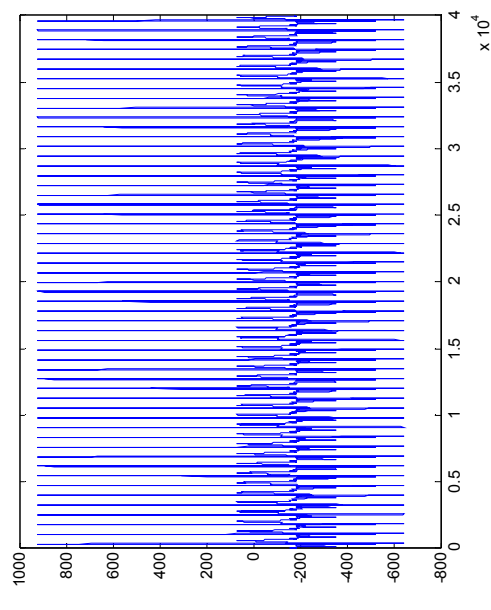
derivación z



derivación x

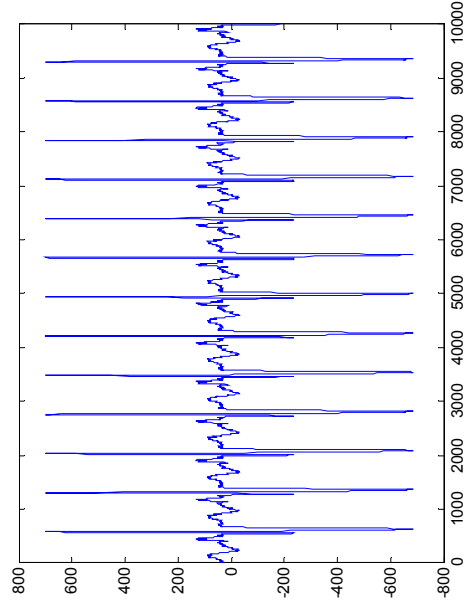


derivación y

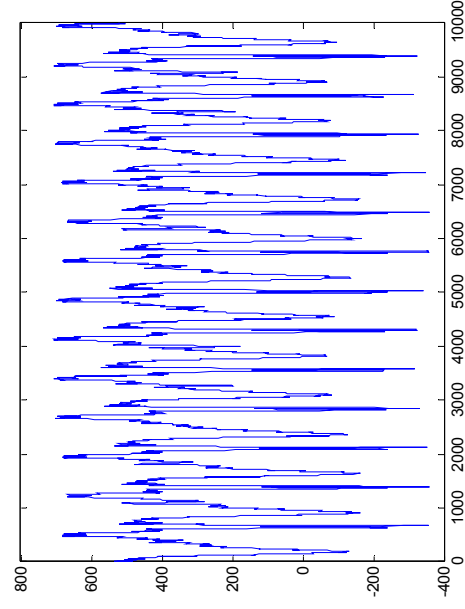


derivación z

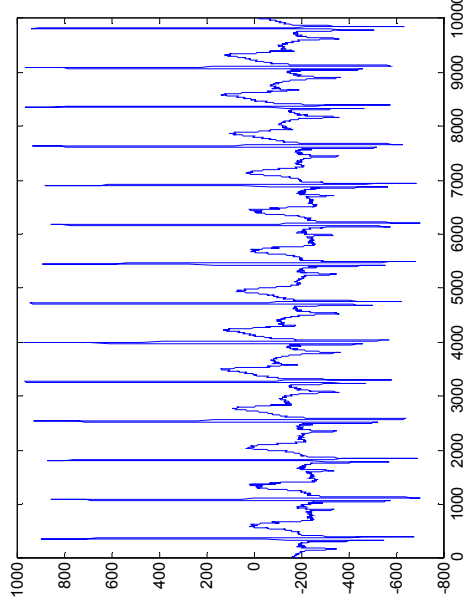
Giro $x=10$ $y=0$, $z=0$



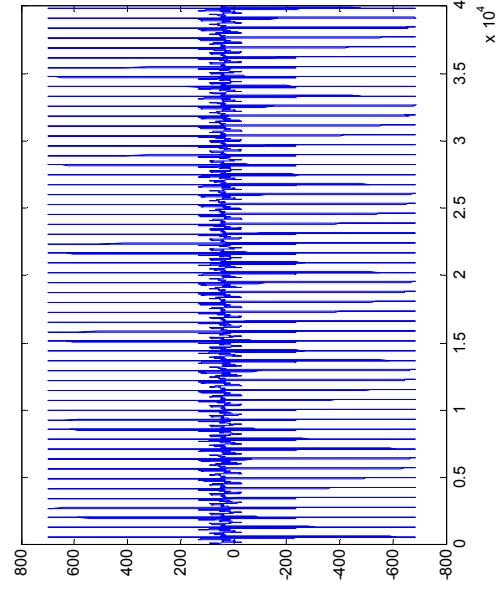
derivación x



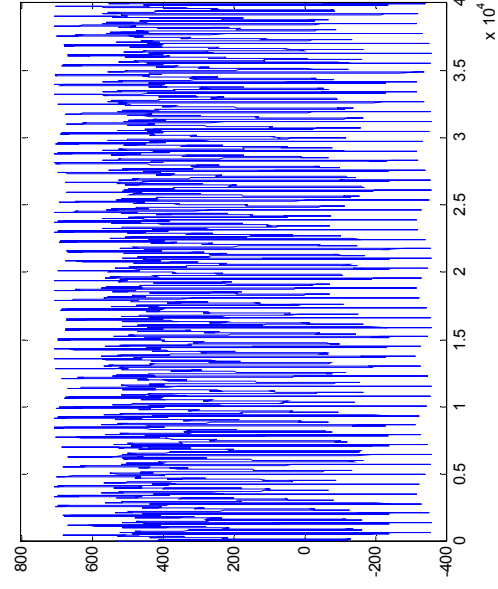
derivación y



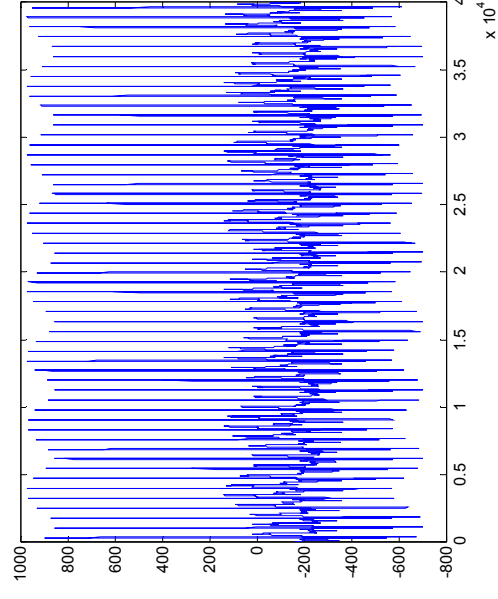
derivación z



derivación x

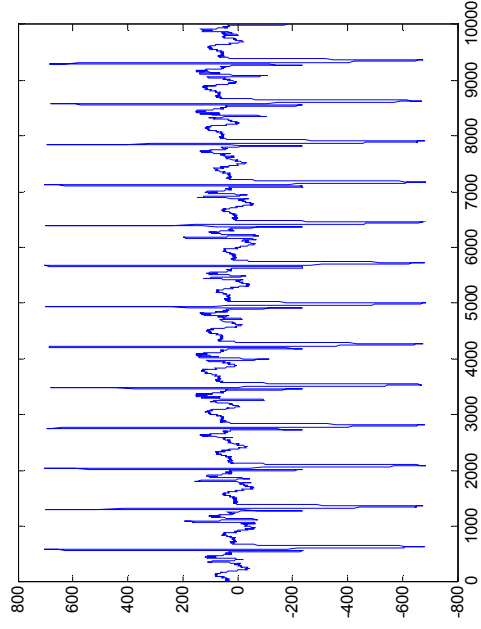


derivación y

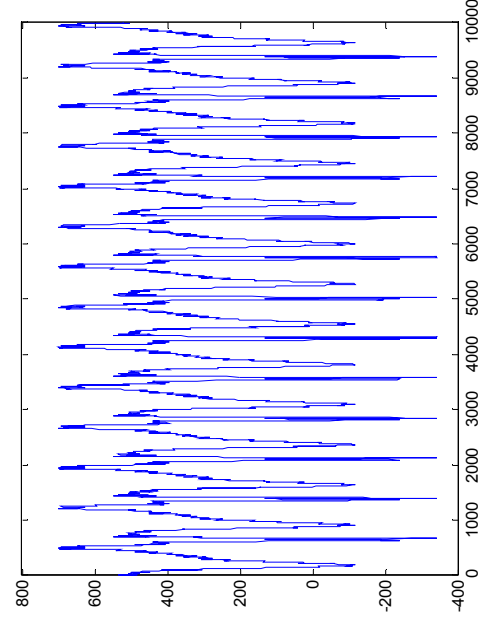


derivación z

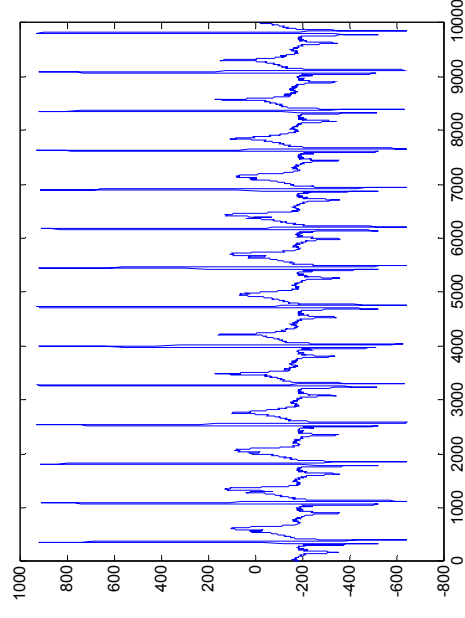
Giro $x=0$ $y=10$, $z=0$



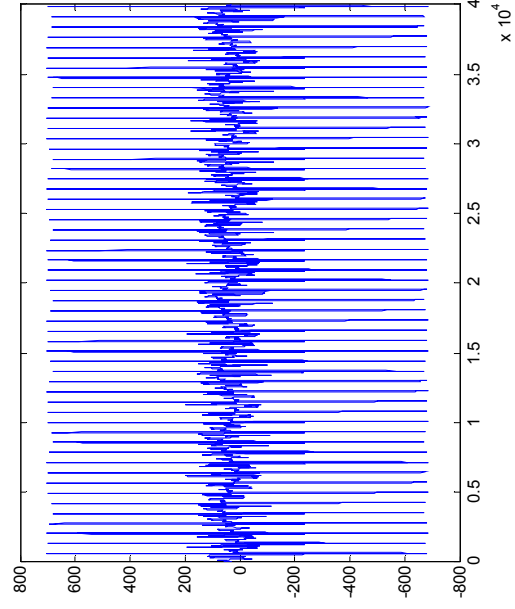
derivación x



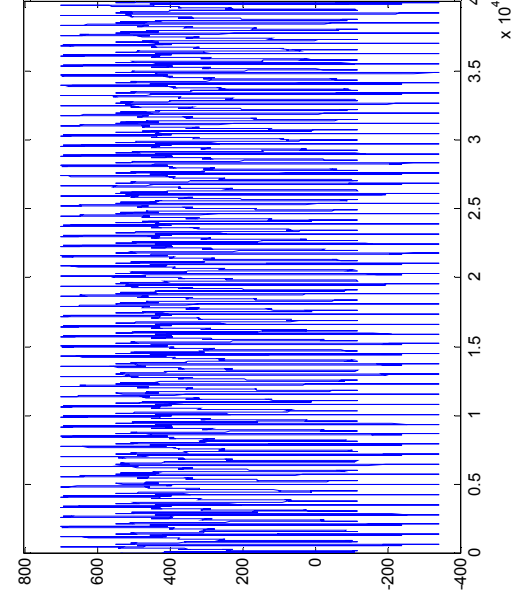
derivación y



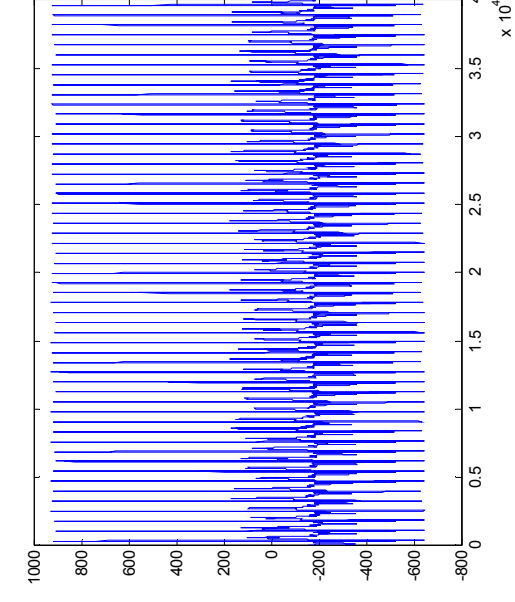
derivación z



derivación x

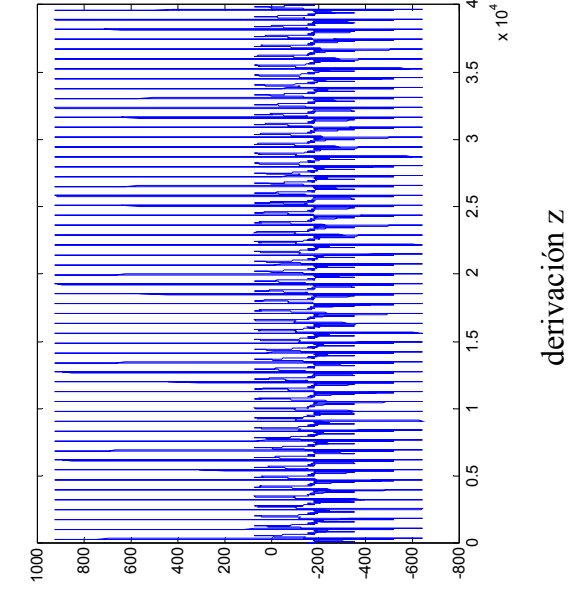
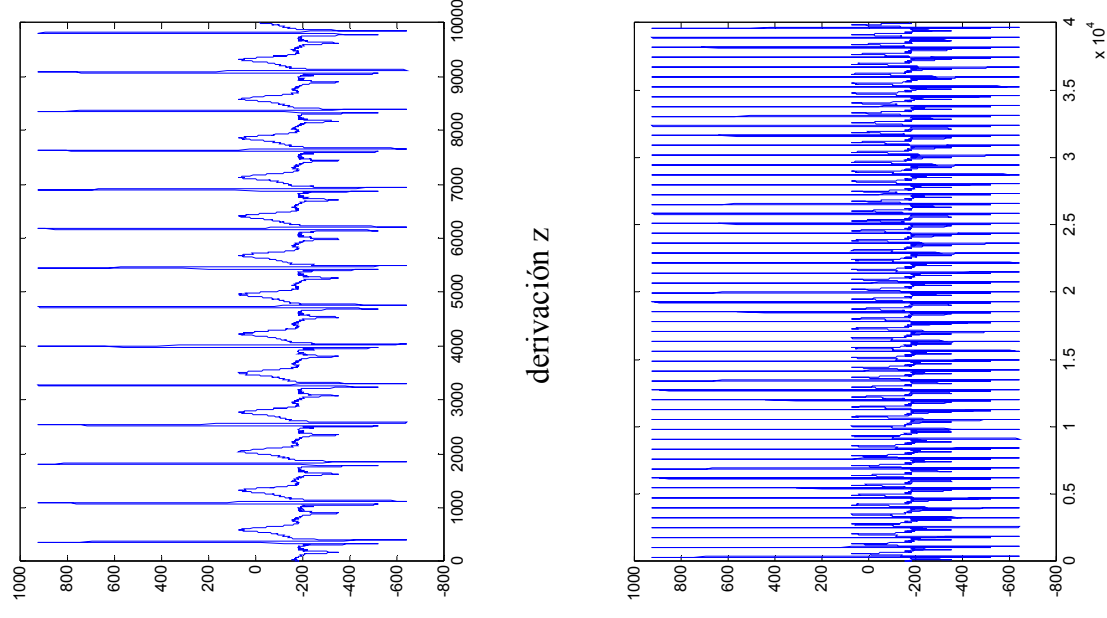
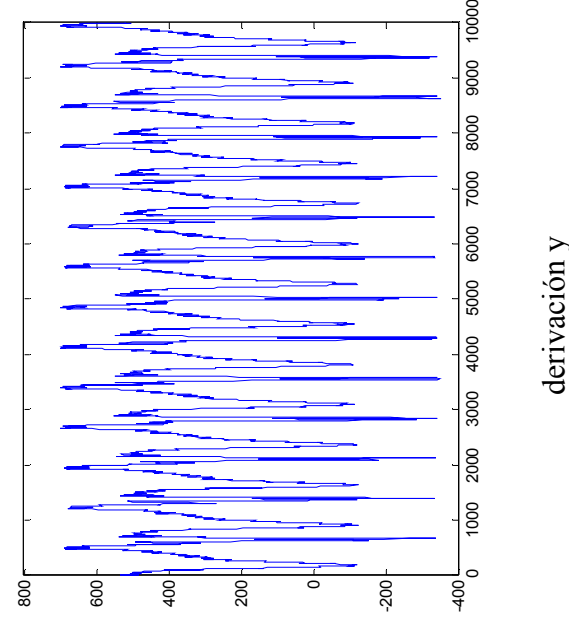
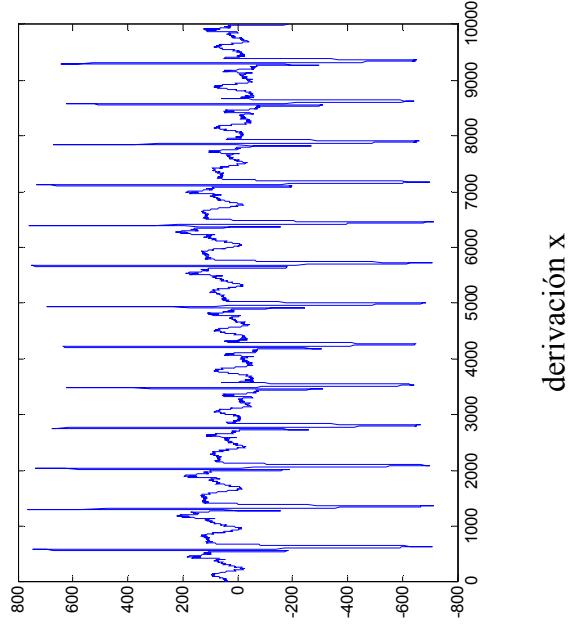


derivación y



derivación z

Giro $x=0$ $y=0$, $z=10$



7. Detección de complejos QRS a partir de ECGs sintéticos rotados

En este apartado, hemos procedido a verificar el funcionamiento de los detectores cuando se les introduce una señal ECG modificada artificialmente como las del apartado anterior. Para ello, en primer lugar, hemos limitado el caso de estudio a los diez primeros pacientes del banco. Debido al hecho de que la detección de los complejos QRS se puede realizar a partir de cualquiera de las tres derivaciones, hemos optado por coger solamente las derivaciones “x”. De esta manera, restringimos la casuística de trabajo y, al mismo tiempo, evitamos operar con derivaciones como la “y” donde generalmente la detección de los picos QRS será hartó más complicada. A continuación, extraeremos las series RR de la que calcularemos el parámetro SDRR o desviación estándar de la serie RR que nos permitirá cuantificar la calidad de la detección.

Los resultados que aparecen en las siguientes tablas son la media de las SDRR, en milisegundos, dado un detector y dado un paciente para todos los 14 giros apuntados en el apartado anterior.

detector	paciente 1	paciente 2	paciente 3	paciente 4	paciente 5
wavelet	0	0,15257	0	0	0
GA	0	0	0	0,46627	0
Hilbert	0,32309	0	0,22133	0,31132	0
zero crossings	0,32293	0	0,22139	0,31111	0
length	12,848	0,48488	29,132	2,9565	5,3834
energy	0,25799	3,7676	0	0,24655	0,44101
hamilton	0,23009	2,6773	0,056377	0,41013	0,37124
ruha	32,15	0	0,41487	0	7,5696
trahancias	54,033	105,7	52,711	6,2305	6,1122

detector	paciente 6	paciente 7	paciente 8	paciente 9	paciente 10
wavelet	0,41735	0,16479	0	0,45157	0,17784
GA	0,075262	0,06356	0	0,075262	0,18121
Hilbert	0,65895	0	0	0,067205	6,2827
zero crossings	0,65976	0	0	0,067137	6,2841
length	0,78808	1,5639	13,89	3,4818	8,7105
energy	0,43703	2,9204	0,21561	0,17306	2,6466
hamilton	0,18878	0,69127	0,01292	0,14715	1,9972
ruha	0,40962	16,141	23,291	10,853	44,178
trahancias	29,008	6,6848	7,6054	40,562	77,938

Tabla 1. SDRR (ms) en la detección del complejo QRS para cada paciente y para cada detector.

Debido al hecho de que el error inducido por el detector de Trahanias en todos y cada uno de los pacientes es mucho mayor que los demás hemos decidido no incluirlo en el apartado de estadísticas. La tabla 2 nos muestra la variación de las SDRR medias para los diferentes detectores (excluyendo Trahanias). Además, nos hacen vislumbrar que posiblemente haya pacientes donde el cambio de morfología en el ECG es mayor que otros. En nuestro caso, el paciente 10 es el que presenta más cambios y, por el lado contrario, el paciente 6 es el que presenta menos.

	paciente 1	paciente 2	paciente 3	paciente 4	paciente 5
Máximo	32,15	3,7676	29,132	2,9565	6,1122
Mínimo	0	0	0	0	0
Media	5,76651	0,88529	3,75575	0,58774	1,72066

	paciente 6	paciente 7	paciente 8	paciente 9	paciente 10
Máximo	0,78808	6,6848	13,89	10,853	44,178
Mínimo	0,075262	0	0	0,067137	0,17784
Media	0,36348	1,20887	2,17239	1,53162	7,04582

Tabla 2. SDRR (ms) máxima, mínima y media para cada paciente.

Con todo, obtenemos las siguientes medias de desviaciones estándar para cada detector:

detector	SDRR media
wavelet	0,13641
GA	0,086156
Hilbert	0,78646
zero crossing	0,78643
Length	7,924
energy	1,1106
hamilton	0,67824
ruha	13,501
trahanias	38,658

Tabla 3. SDRR media (ms) para cada detector

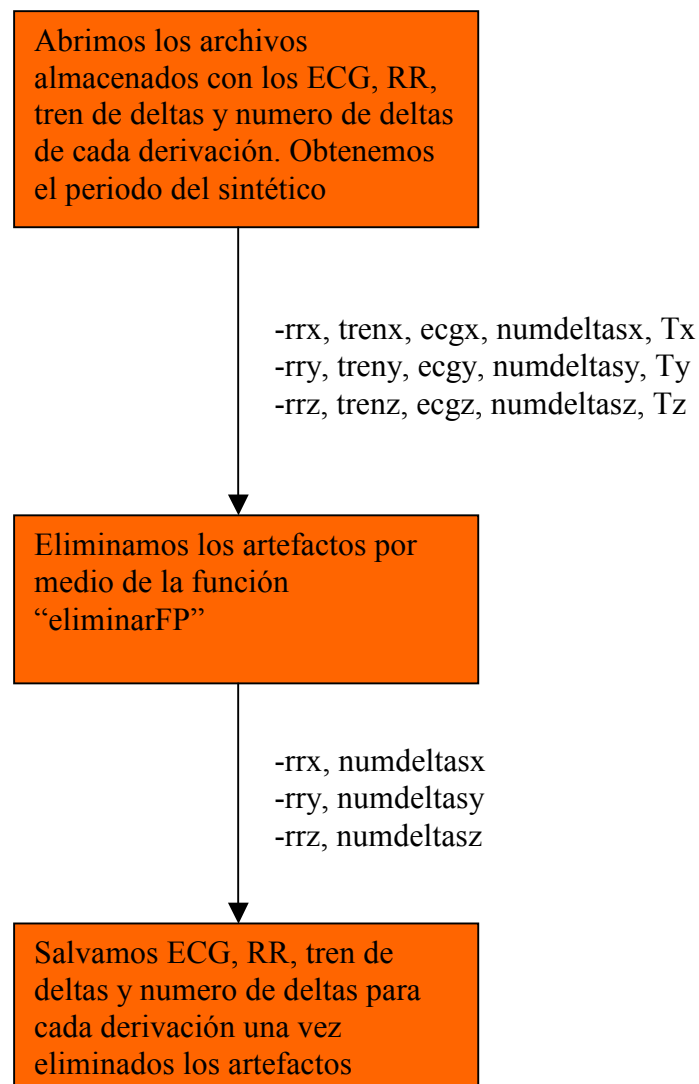
Estos resultados no son del todo concluyentes en cuanto a sensibilidad a cambios de morfología puesto que incluyen errores aberrantes debidas a detecciones de falsos positivos y falsos negativos. Sin embargo, ya se puede hacer una primera idea de que

detectores tienen un mejor comportamiento (como es el caso del Genetic Algorithm y Wavelet) y de los que funcionan peor (principalmente Ruha y Trahanias).

8. Eliminación de artefactos tras la detección

Una vez realizada la primera detección procederemos a eliminar los falsos negativos y los falsos positivos con el fin de que los errores aberrantes no afecten a la SDRR ni a la corrección que posteriormente llevaremos a cabo.

El algoritmo empleado se llama “precorreccionRRspararesults” y su funcionamiento es el siguiente:



La función clave de la eliminación de artefactos es “eliminarFP” y su desarrollo es el que sigue:

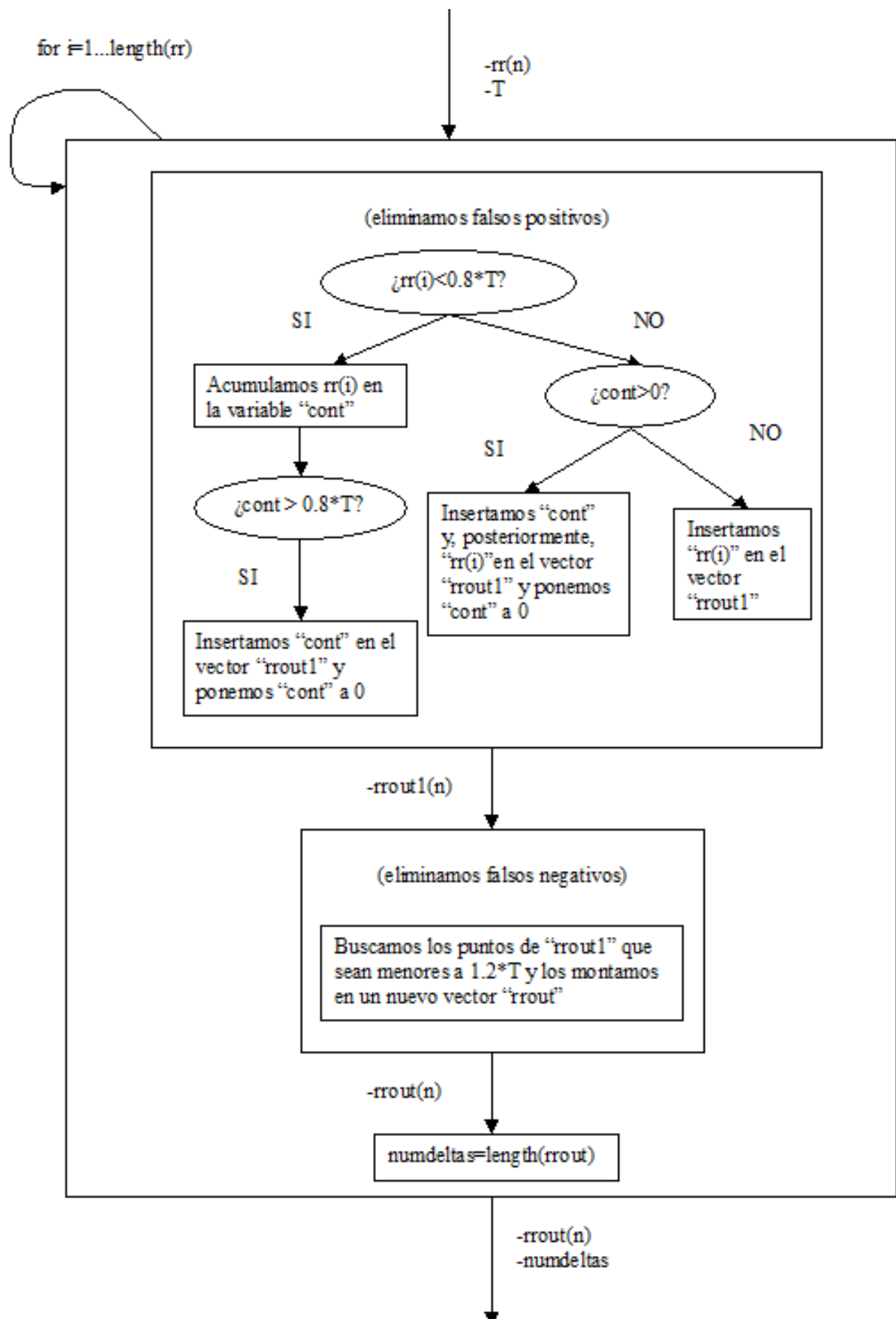


Fig. 61 Flujograma de la función "eliminarFP".

En las siguientes dos figuras se puede comparar la detección antes y después de la corrección de artefactos.

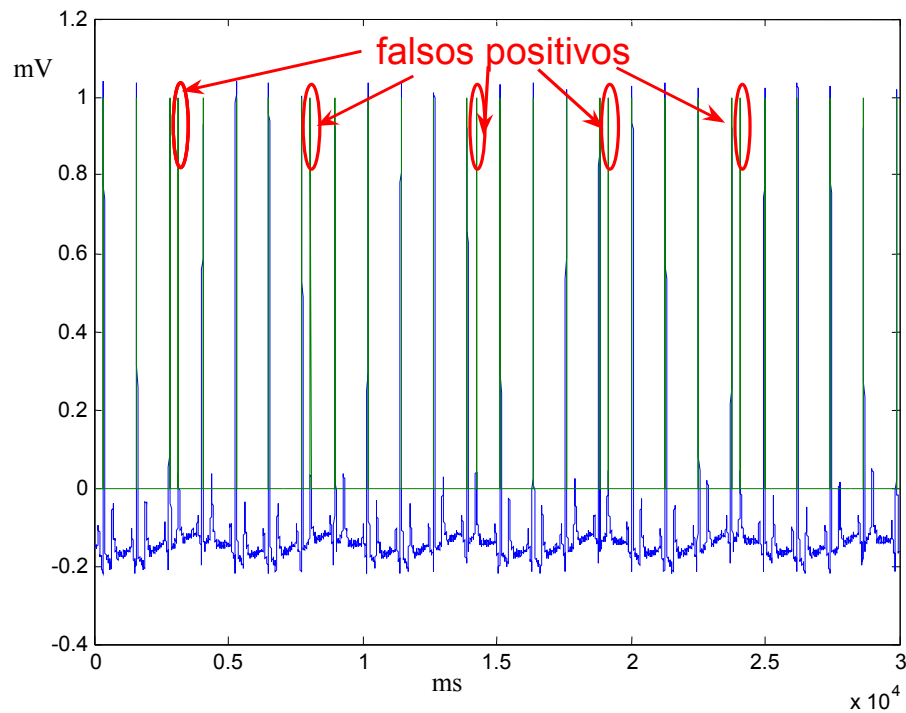


Fig. 62 Tren de deltas superpuesto al ECG tras la detección por medio del detector Ruha.

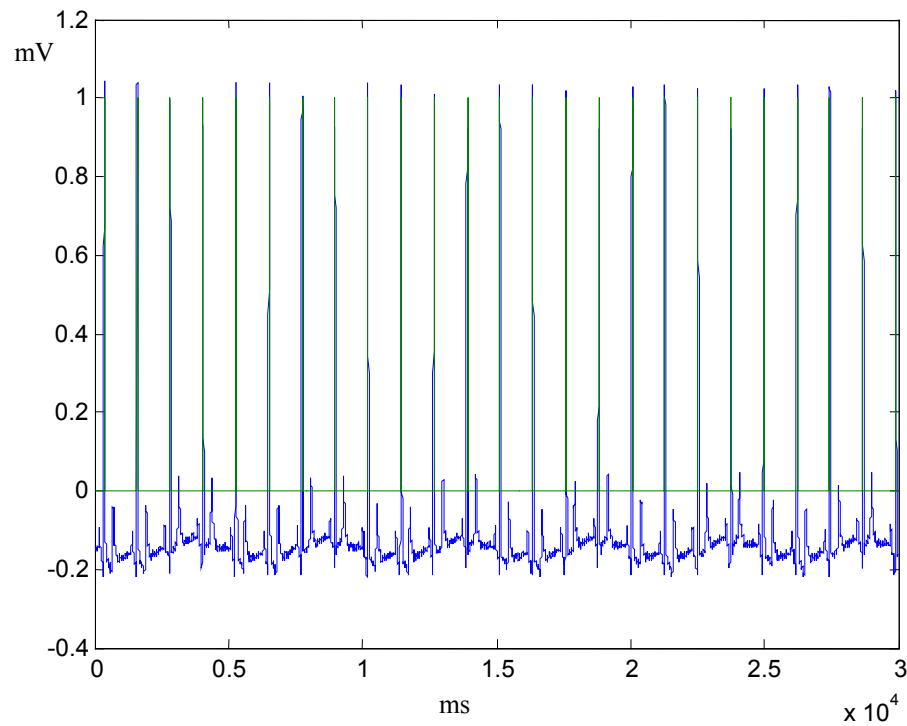


Fig. 63 Tren de deltas superpuesto al ECG tras la detección por medio del detector Ruha tras la eliminación de artefactos.

Además, dado que en la práctica siempre dispondremos de las tres derivaciones hemos escogido la derivación en la que la SDRR es mínima. Bajo estas premisas hemos obtenido los siguientes resultados:

detector	paciente 1	paciente 2	paciente 3	paciente 4	paciente 5
wavelet	0	0,076438	0	0	0
GA	0	0	0	0,077791	0
Hilbert	0,32343	0	0,11056	0,15577	0
zero crossings	0,23309	0	0,1506	0,18588	0
length	8,3289	0,40864	0	1,6052	5,3171
energy	0,25799	1,0835	0	0,20761	0,21181
hamilton	0,23009	0,84496	0,056377	0,33226	0,22657
ruha	19,462	0	0,41494	0	7,3636
trahanias	37,343	21,787	17,924	5,9414	5,975

detector	paciente 6	paciente 7	paciente 8	paciente 9	paciente 10
wavelet	0,067832	0,12334	0	0,37641	0,073658
GA	0	0	0	0,07506	0,055304
Hilbert	0	0	0	0	0
zero crossings	0	0	0	0,26683	0
length	0,66547	1,5639	9,5837	1,262	6,7576
energy	0,11694	2,1575	0,12429	0,17306	0,012506
hamilton	0,14375	0,59191	0,01292	0,14715	0,22898
ruha	0,40968	2,5167	1,3801	8,6982	0,25149
trahanias	26,866	6,6848	7,6054	11,815	8,9772

Tabla 4. SDRR (ms) para cada paciente con cada detector tras la eliminación de artefactos.

En las siguientes tablas se puede apreciar que, por medio de la elección de la derivación más óptima y gracias a la eliminación de los artefactos, la media de error se ha reducido considerablemente así como los máximos.

	paciente 1	paciente 2	paciente 3	paciente 4	paciente 5
Máximo	19,462	1,0835	0,41494	1,6052	7,3636
Mínimo	0	0	0	0	0
Media	3,61569	0,30169	0,08657	0,31681	1,63989

	paciente 6	paciente 7	paciente 8	paciente 9	paciente 10
Máximo	0,66547	2,5167	9,5837	8,6982	6,7576
Mínimo	0	0	0	0	0
Media	0,17546	0,86917	1,38763	1,34149	0,92244

Tabla 5. SDRR (ms) máxima, mínima y media para cada paciente tras la eliminación de artefactos.

Así pues, las SDRR medias resultantes para cada detector figuran en la siguiente tabla:

detector	SDRR media
GA	0.020816
Wavelet	0.071768
Hamilton	0.2815
Zero crossing	0.35896
Hilbert	0.37897
Energy	0.43452
Length	3.5492
Ruha	4.0497
Trahanias	15.092

Tabla 6. SDRR (en ms) media para cada detector tras la eliminación de artefactos.

Como se puede ver, los detectores que han presentado mayor robustez frente a cambios de morfología del ECG son Wavelet, Genetic Algorithm, Hilbert, Zero Crossing, Energy y Hamilton. Los más sensibles son Length y Ruha. Para los posteriores estudios descartaremos el detector Trahanias debido a que el error que produce es mucho mayor que el resto y, en consecuencia, se lo considerará como un error aberrante. Así pues, escogeremos el mejor y el peor detector para las siguientes etapas de nuestro estudio que son Genetic Algorithm y Ruha respectivamente.

9. Limitación de la casuística

Una vez pasados los diez primeros pacientes de la base de datos por los detectores, nuestra intención será la de acotar un poco los casos de estudio. Para ello, hemos realizado un promediado de las desviaciones estándar en función del giro aplicado con la finalidad de buscar si existe algún giro crítico. Seguidamente podemos ver una tabla con los valores medios de las desviaciones de las series RR obtenidas para cada detector en función del giro aplicado para el paciente 1 (los giros han sido aplicados simultáneamente en las tres derivaciones del complejo ECG para cada ensayo):

giro	energy	GA	hamilton	hilbert	length	ruha	trahanas	wavelet	zero crossing
2,5°	0,045083	0	0,0448	0,53912	0,045083	0,5466	53,398	0,53991	0,53991
5,0°	12,805	0	0,5326	0,54045	0,13537	0,78476	62,645	0,53991	0,53991
7,5°	12,837	0	0,5401	0,7276	1,3471	3,5738	171,69	0,76354	0,72826
10,0°	12,998	0	0,5401	0,77877	1,0142	3,7493	193,32	0,80735	0,77929
12,5°	13,11	0,53991	0,5401	0,85334	0,90933	5,1909	196,97	1,1989	0,85367
15,0°	13,921	0,53991	0,5401	0,90188	0,90146	11,612	200,15	1,3362	0,90209
17,5°	14,49	0,53991	0,5401	0,93066	3,5411	34,709	185,41	1,6853	0,92971
20,0°	14,932	0,53991	36,2811	1,0368	4,8957	44,186	180,72	2,0623	1,0368
22,5°	15,208	0,53991	90,8440	1,1605	8,24	44,098	183,38	2,7156	1,1602
25,0°	15,417	0,53991	93,5086	1,6338	145,75	42,765	188,45	3,5541	1,6377
27,5°	16,034	4,4803	94,8842	1,9023	157,9	54,322	194,59	4,7381	1,9051
30,0°	16,817	155,72	93,2422	5,9477	178,27	72,159	190,72	5,8716	5,9662

Tabla 7. SDRR (en ms) para cada detector en función del giro aplicado

El comportamiento observado en los resultados es coherente con lo esperado ya que, en general, la SDRR aumenta con la amplitud del giro aplicado. La mayoría de detectores se mantienen estables para un cierto rango de giros y a partir de un valor determinado la desviación estándar de su RR se dispara. Según la robustez del detector frente a los giros, dicho valor se disparará antes o después.

Dado que en la práctica haremos uso de una de las tres derivaciones para llevar a cabo la detección, aplicaremos la misma amplitud a los tres casos con el fin de que se afecte por igual a las tres.

Para nuestro posterior estudio hemos escogido dos casos relevantes que serían el giro de 2,5° y el giro de 12,5°. El primero representaría los casos en los que la afectación de la respiración sobre la morfología del ECG sea leve. Por el contrario, el segundo caso

se correspondería con un caso extremo al que probablemente nunca se llegaría, por ejemplo, en el caso de rotaciones producidas de la respiración. Hemos escogido este giro de $12,5^\circ$ para reflejar una situación en la que la SDRR de los detectores se dispare y así poder comparar el rendimiento de los detectores en ambos casos (uno donde los cambios de morfología sean leves y otros donde sean muy fuertes).

Asimismo, a raíz de los resultados anteriores, se escogerá el detector más robusto (Genetic Algorithm) y el detector menos robusto (Ruha) con el fin de que haya más contraste en posteriores estudios.

Una vez restringidos el número de detectores y el número de giros que emplearemos procederemos a trabajar con todos los pacientes de la base de datos.

10. Estudio sobre todos los pacientes de la base de datos¹²

En primer lugar, se ha obtenido los ECGs sintéticos para los 294 pacientes de la base de datos. En segundo lugar, se les ha sometido a los giros escogidos anteriormente de $2,5^\circ$ y de $12,5^\circ$ para las tres derivaciones. A continuación, hemos pasado los ECGs sintéticos rotados por los dos detectores escogidos. Como en la práctica siempre se optará por escoger como SDRR la de la derivación donde se dé la desviación estándar menor (por ser la que mejor relación señal a ruido tiene), se ha procedido a seguir dicha directiva en la elaboración de las tablas. Por último, se ha procedido a la eliminación de posibles artefactos (es decir, tanto falsos positivos como falsos negativos) valorando como máxima que haya una delta por cada latido del ECG sintético (en nuestro caso, 500 latidos).

Dado que había algunos números que no tenían un paciente asociado a ellos no han sido tenidos en cuenta a la hora de generar las gráficas.

Asimismo, se ha detectado la presencia de pacientes cuyos ECGs presentaban un nivel de ruido exagerado que alteraría los resultados del estudio. Un ejemplo podría ser el del paciente 285 que tenía el siguiente ECG (que se ha descartado para el estudio):

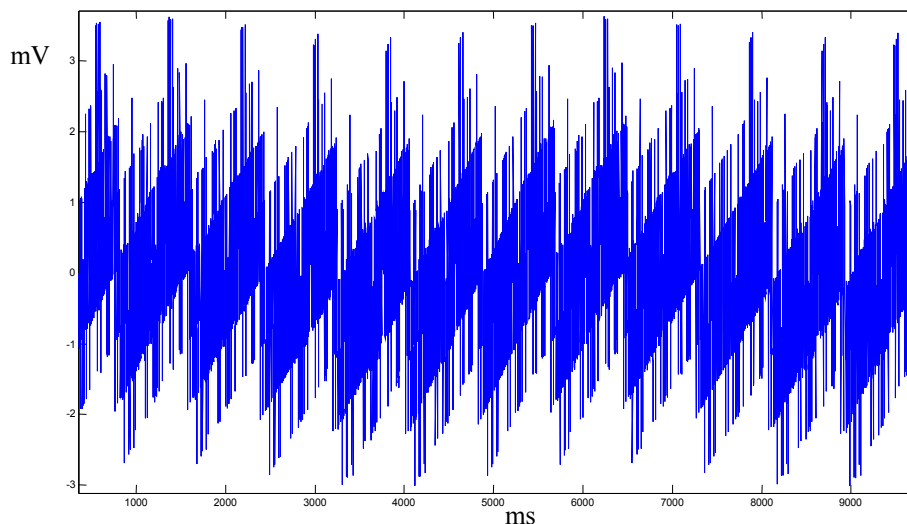


Fig. 64 Señal ECG rotada $2,5^\circ$ del paciente número 285

A parte del paciente 285 se ha encontrado casos similares en los pacientes 111, 286 y 292, que también han sido considerados como casos aberrantes y, por lo tanto, excluidos del estudio.

Las gráficas para el detector Genetic Algorithm quedan de la siguiente manera:

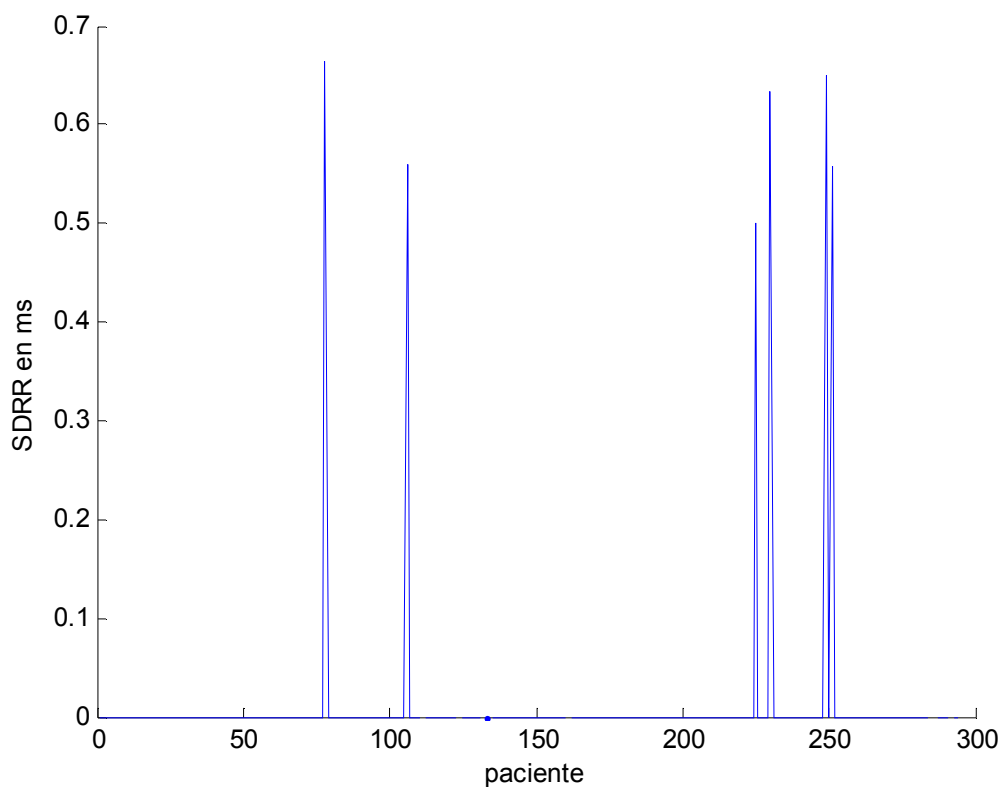


Fig. 65 SDRR (ms) en función del paciente de estudio para el giro de 2,5°

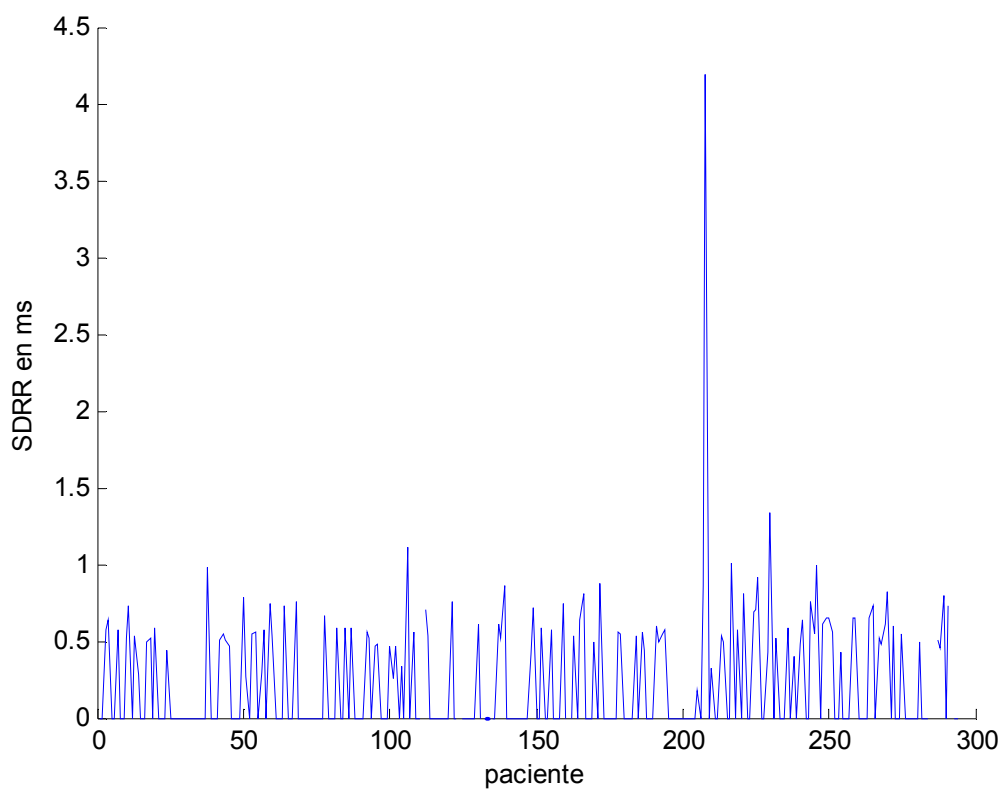


Fig. 66 SDRR (ms) en función del paciente de estudio para el giro de 12,5°

Los valores más significativos obtenidos en los dos estudios realizados por medio del detector Genetic Algorithm son los siguientes:

	giro 2,5°	giro 12,5°
media	0.0125	0.2393
máximo	0.6645	4.1852
mínimo	0	0

Tabla 8. SDRR (ms) media, máxima y mínima para el detector Genetic Algorithm.

Los resultados obtenidos para el detector de Ruha son:

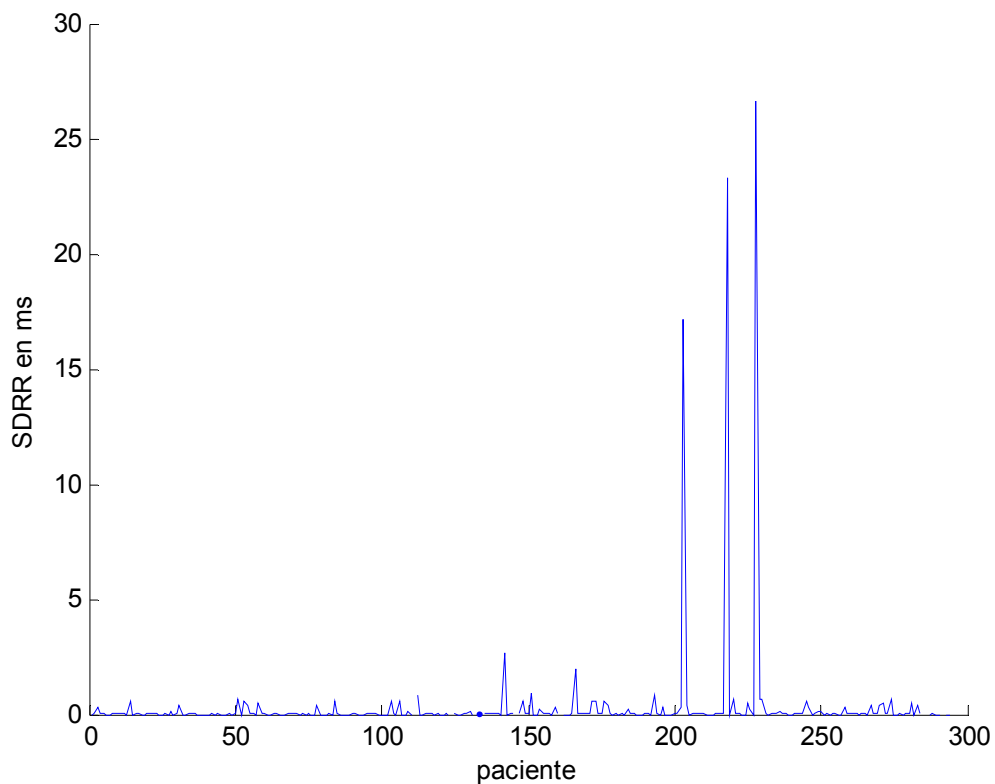


Fig. 67 SDRR (ms) en función del paciente de estudio para el giro de 2,5°

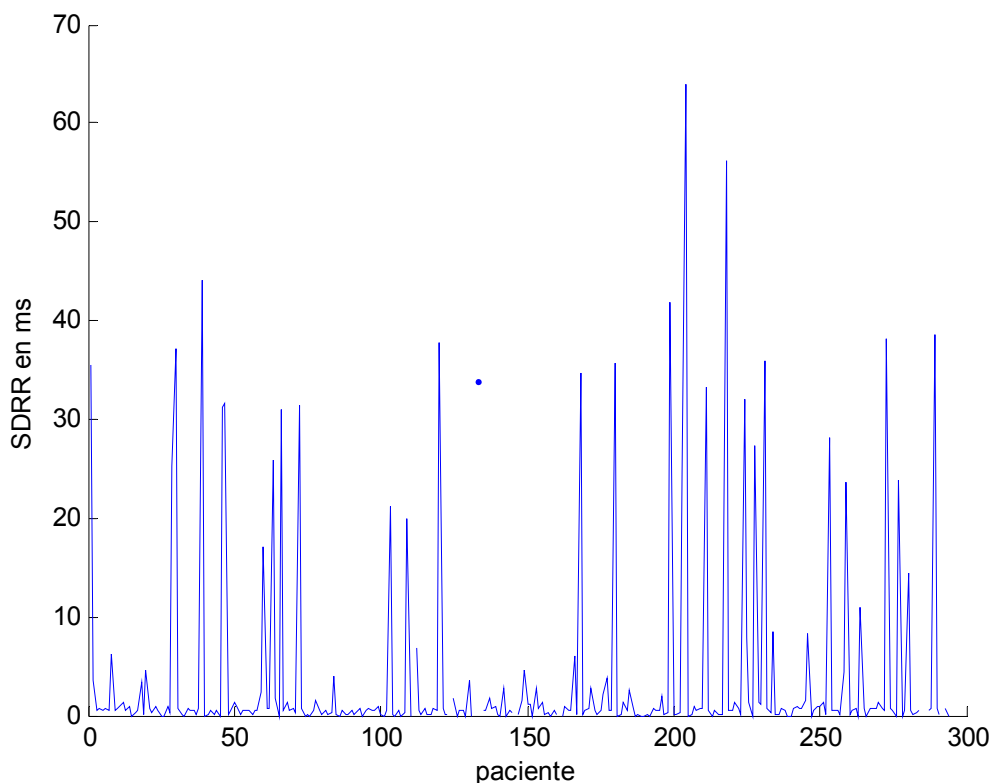


Fig. 68 SDRR (ms) en función del paciente de estudio para el giro de 12,5°

Los valores más significativos obtenidos para el detector de Ruha son:

	giro 2,5°	giro 12,5°
media	0.3389	4.1457
máximo	26.6266	64.0639
mínimo	0	0

Tabla 8. SDRR (ms) media, máxima y mínima para el detector Ruha.

Con todo, de este apartado podemos extraer que el detector Genetic Algorithm se presenta bastante robusto tanto para giros pequeños como para giros exagerados como el de 12,5°. La SDRR media para giros de 2,5° es de 0.0125 ms y para giros de 12,5° es de 0.2393 ms. Por el contrario, el detector Ruha se muestra más sensible a los cambios de morfología introducidos obteniendo una media de 0.3389 ms de SDRR para el giro pequeño y una media de 4.1457 ms para el giro de 12,5°.

11. Corrección de las detecciones mediante el detector por correlación

Es muy común tras una primera detección de QRS afinarla mediante un detector de correlación. Éste último, es considerado como el mejor detector y, por ello, será empleado a modo de referencia. El objetivo de este capítulo es ver si tras esta corrección aumenta la sensibilidad del detector frente a cambios en la morfología del ECG para, de esta manera, determinar la conveniencia de incluir una etapa de corrección. Así pues, los resultados obtenidos tanto por parte del detector “Ruha” como por parte del detector “Genetic Algorithm” se compararán con los resultados derivados de la detección mediante la correlación. Aquellos puntos en los que no coincidan se corregirá aplicando el valor obtenido por el algoritmo de la correlación. Partiendo de la base de que la variabilidad de ritmo cardíaco que presentan los ECGs sintéticos debería ser nula podremos deducir si la SDRR debida exclusivamente a cambios de morfología es cancelable o no por medio de un corrector.

Un parámetro relevante en la corrección por correlación es el valor asignado a la ventana. Dicha ventana es la mitad del intervalo temporal por el cual se conforma la máscara del pulso base del ECG con el que se procederá a correlar uno a uno los distintos latidos. Así pues, cuanto mayor sea la ventana, más se aproximará la máscara al pulso y mayor precisión tendrá la corrección (no obstante, el coste computacional será mayor y, además, el valor de dicha ventana no puede ser infinitamente grande ya que de lo contrario abarcaría más de un latido y la correlación se realizaría con dos pulsos).

En primer lugar, se realizará un prefiltrado Butterworth de segundo orden con frecuencias de corte a 10 y 30 Hz. El algoritmo corrector por correlación está constituido de la siguiente manera:

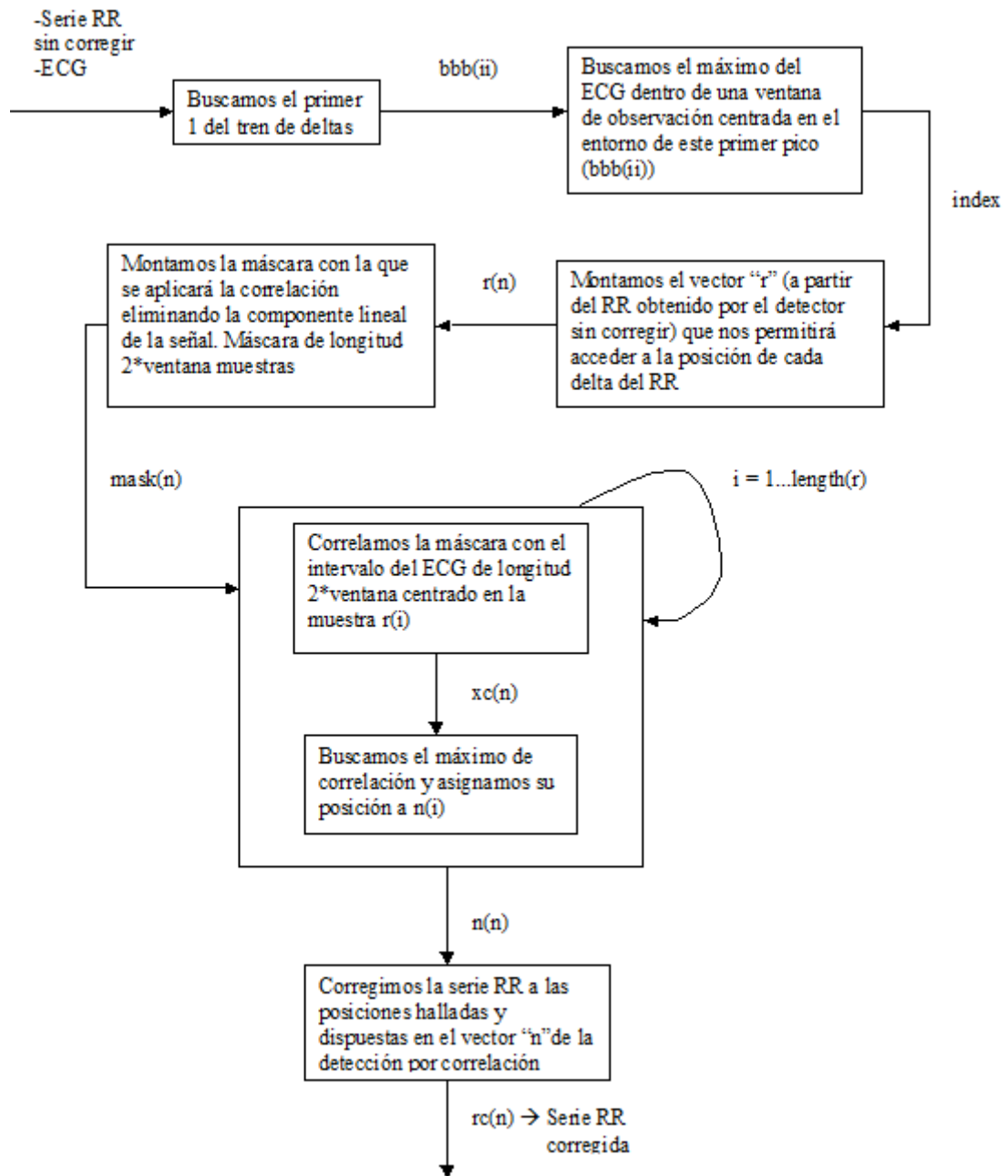


Fig. 69 Flujograma del algoritmo corrector por medio de la correlación

Se ha llevado a cabo un primer estudio que se ha realizado cogiendo de forma arbitraria una ventana de 100 muestras a una frecuencia de muestreo de 1000 Hz (es decir, una ventana de 100 ms).

Los resultados obtenidos por el detector “Genetic Algorithm” una vez corregido por correlación y para una amplitud de 2,5° de la fase senoidal introducida son los que siguen:

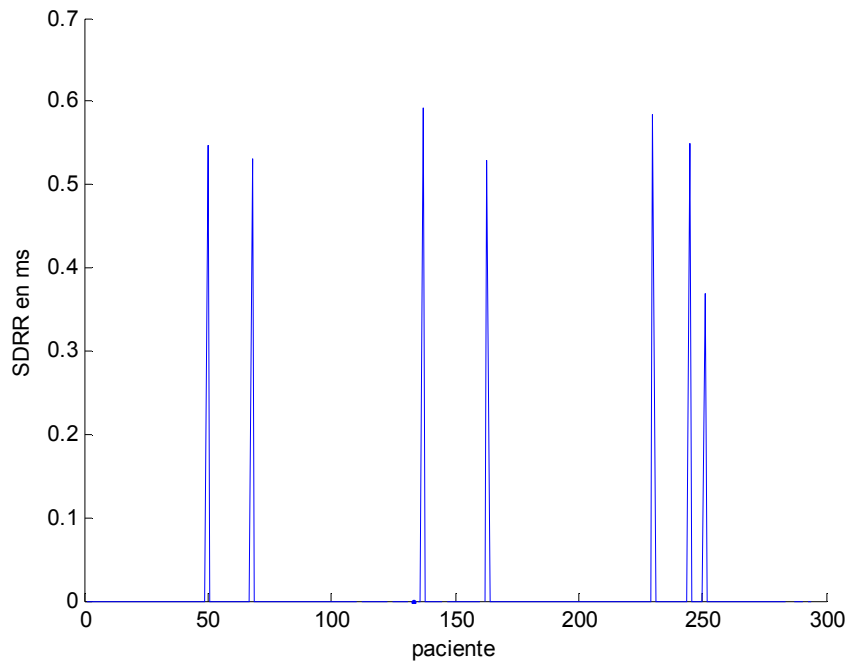


Fig. 70 SDRR (ms) en función del paciente de estudio para el giro de $2,5^\circ$ para el detector Genetic Algorithm.

En la siguiente figura se puede apreciar la señal sin corregir (de color rojo) sobrepuesta a la señal corregida (de color azul):

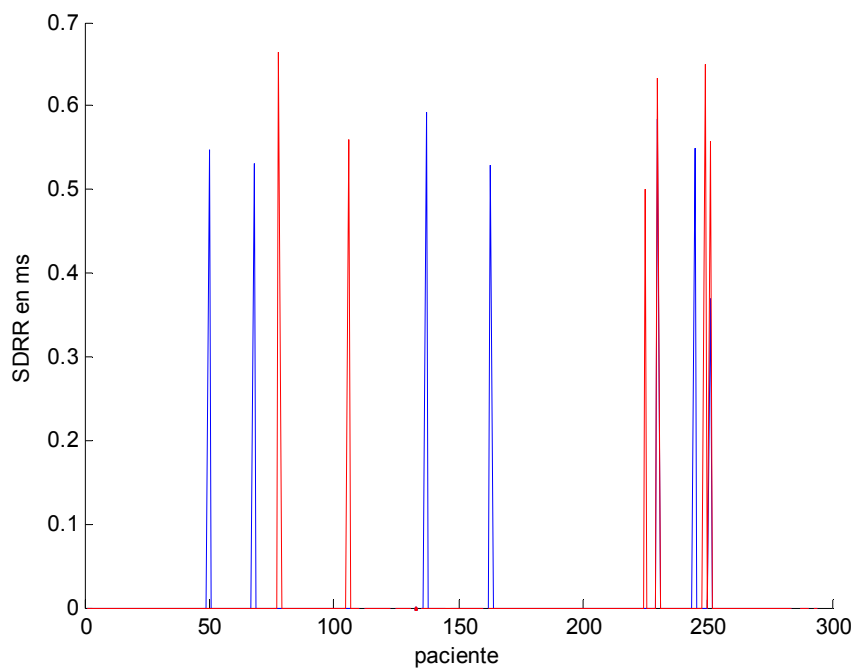


Fig. 71 Comparativa entre señal corregida (azul) y señal sin corregir (rojo) para detector Genetic Algorithm en un giro de $2,5^\circ$.

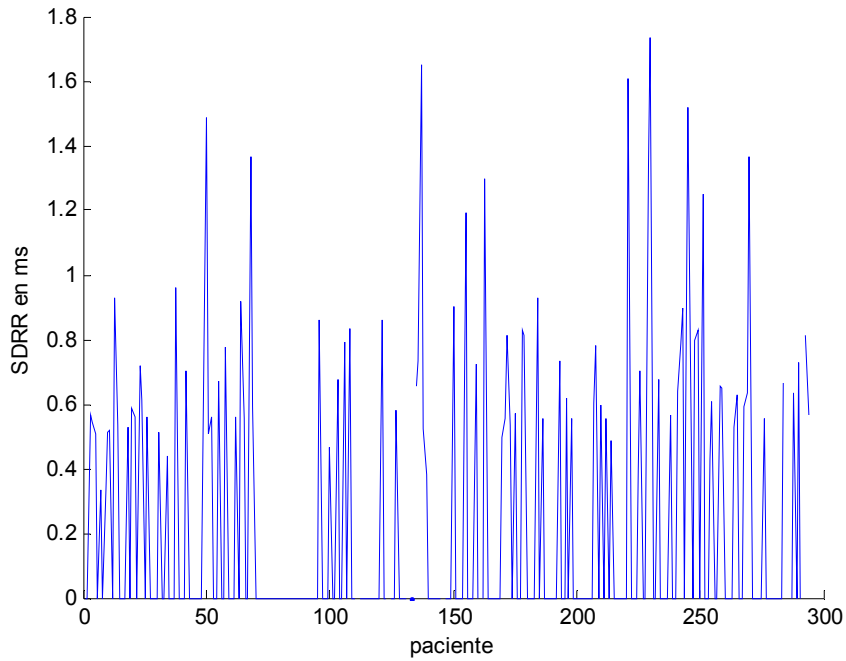


Fig. 72 SDRR (ms) en función del paciente de estudio para el giro de $12,5^\circ$ para el detector Genetic Algorithm.

En la siguiente figura se puede apreciar a la señal sin corregir (de color rojo) sobrepuesta a la señal corregida (de color azul):

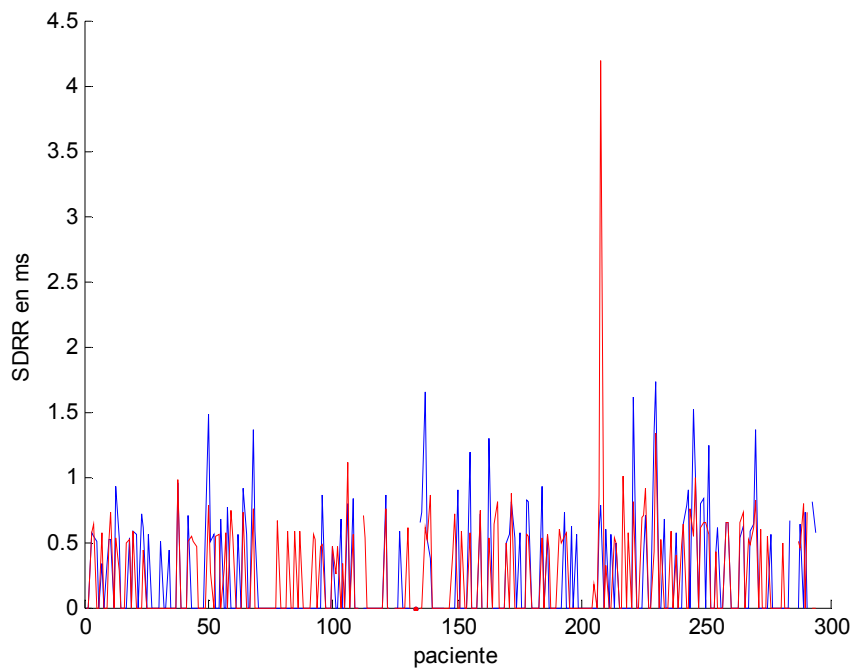


Fig. 73 Comparativa entre señal corregida (azul) y señal sin corregir (rojo) para detector Genetic Algorithm en un giro de $12,5^\circ$.

Los valores más destacables son los siguientes (en rojo tenemos los mismos parámetros obtenidos sin la corrección):

	giro 2,5°	giro 12,5°
media	0.0130	0.2404
máximo	0.5934	1.7361
mínimo	0	0
media	giro 2,5° 0.0125	giro 12,5° 0.2393
máximo	0.6645	4.1852
mínimo	0	0

Tabla 9. SDRR (ms) media, máxima y mínima para el detector Genetic Algorithm con corrección comparadas con los valores obtenidos para el mismo detector pero sin corrección (en rojo).

Los resultados para el detector de Ruha son los siguientes:

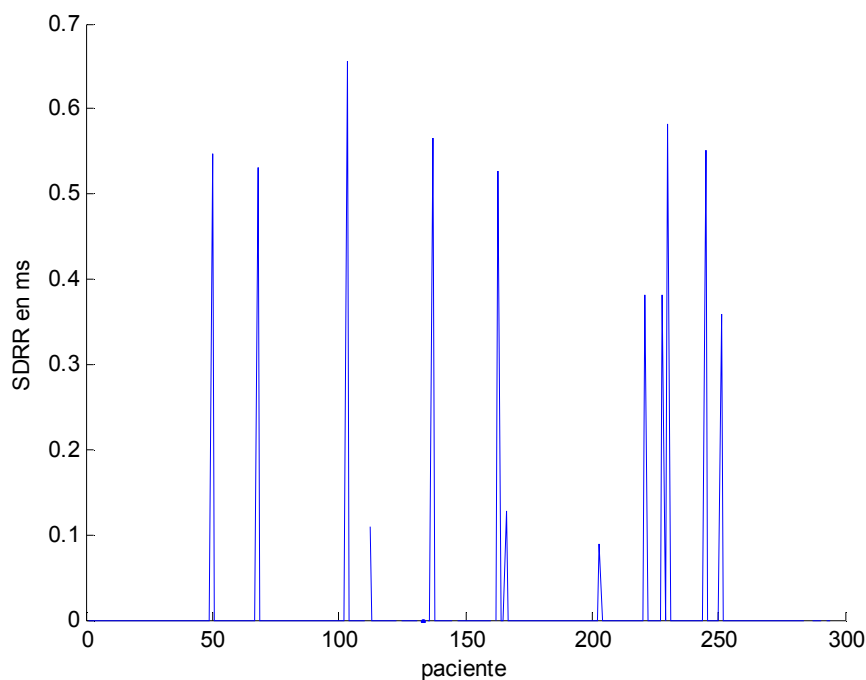


Fig. 74 SDRR (ms) en función del paciente de estudio para el giro de 2,5° para el detector Ruha.

En la siguiente figura se puede apreciar a la señal sin corregir (de color rojo) superpuesta a la señal corregida (de color azul):

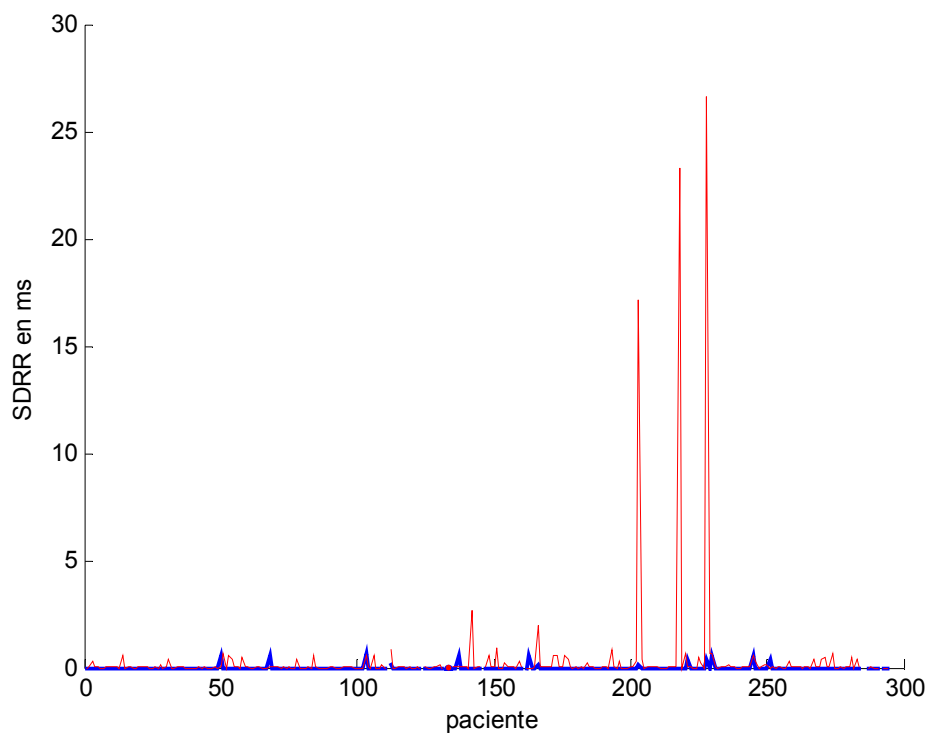


Fig. 75 Comparativa entre señal corregida (azul) y señal sin corregir (rojo) para detector Ruha en un giro de 2,5°.

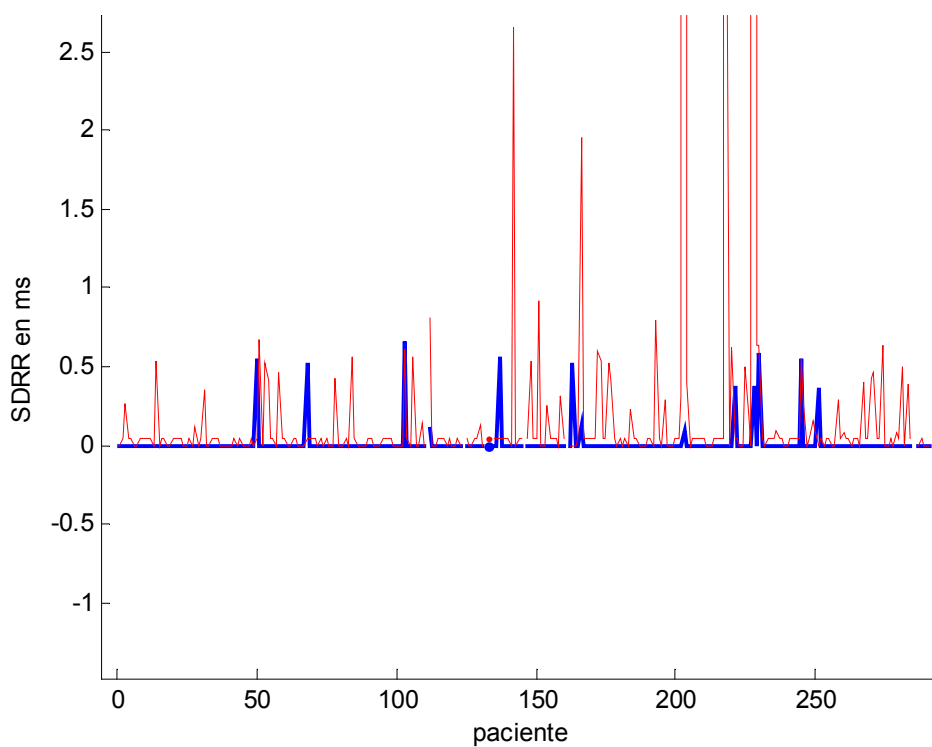


Fig. 76 Zoom de la comparativa entre señal corregida (azul) y señal sin corregir (rojo) para detector Ruha en un giro de 2,5°.

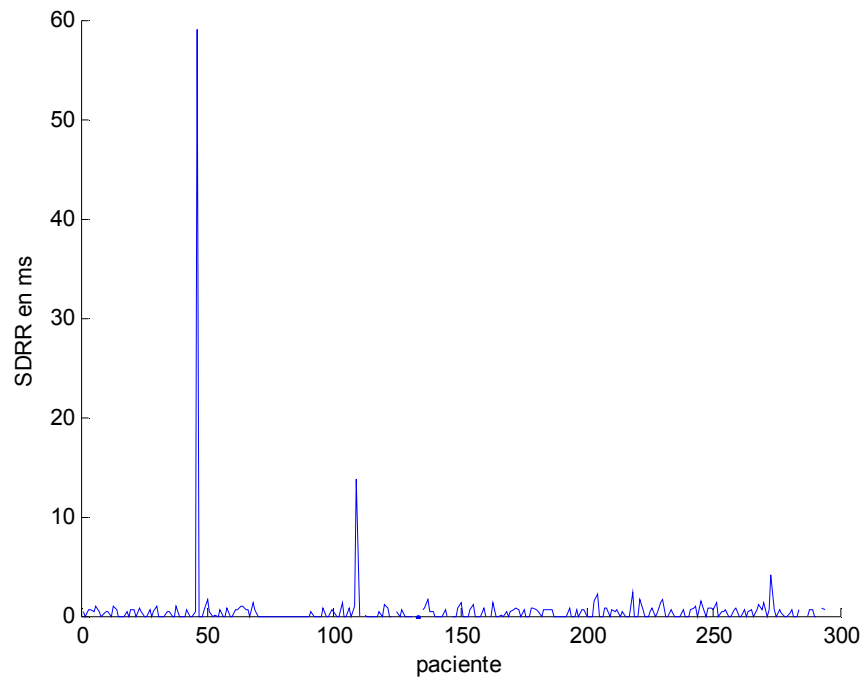


Fig. 77 SDRR (ms) en función del paciente de estudio para el giro de 12,5° para el detector Ruha..

En la siguiente figura se puede apreciar a la señal sin corregir (de color rojo) sobrepuesta a la señal corregida (de color azul):

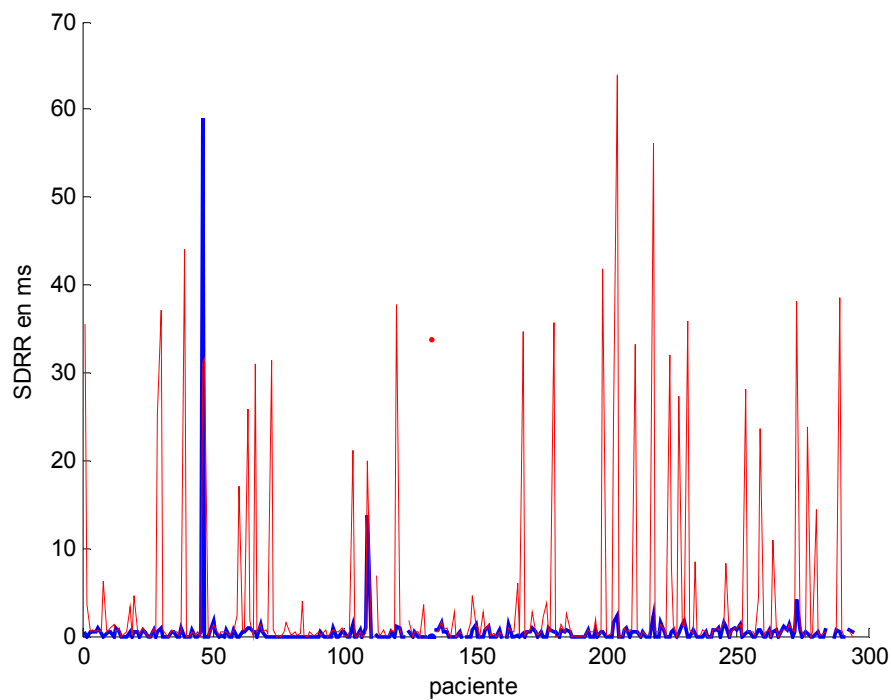


Fig. 78 Comparativa entre señal corregida (azul) y señal sin corregir (rojo) para detector Ruha en un giro de 12,5°.

Los valores más destacables son los siguientes (en rojo tenemos los mismos parámetros obtenidos sin la corrección):

	giro 2,5°	giro 12,5°
media	0.0190	0.6083
máximo	0.6568	59.0122
mínimo	0	0
media	giro 2,5° 0.3389	giro 12,5° 4.1935
máximo	26.6266	64.0639
mínimo	0	0

Tabla 10. SDRR (ms) media, máxima y mínima para el detector Ruha con corrección comparadas con los valores obtenidos para el mismo detector pero sin corrección (en rojo).

A continuación, podemos ver los resultados obtenidos por cada detector variando el parámetro de la ventana.

En primer lugar, se ha probado con el detector Genetic Algorithm para un giro de 2,5°. Las tres siguientes gráficas son para ventanas de 50, 100 y 200 ms.

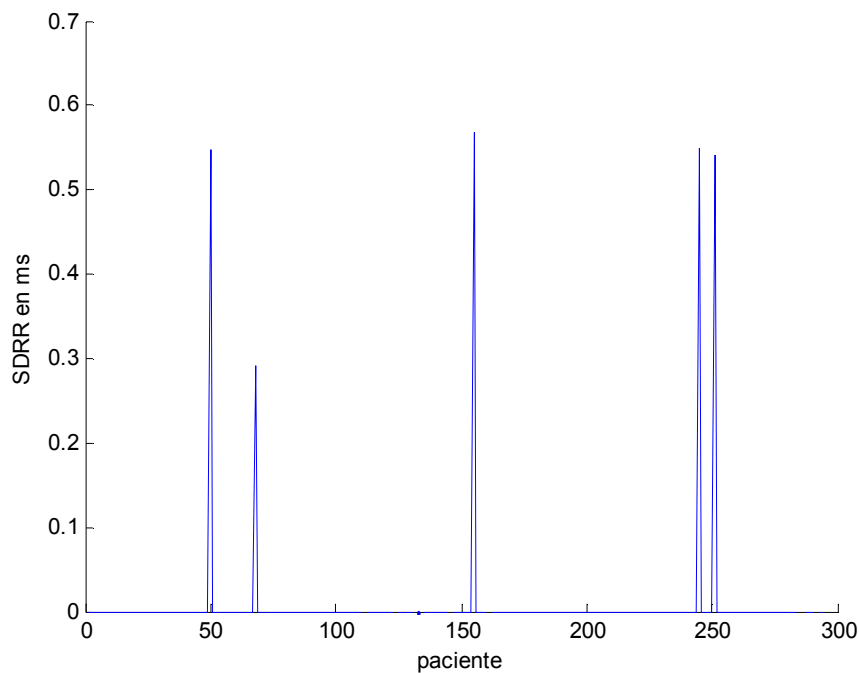


Fig. 79 SDRR en función del paciente para detector Genetic Algorithm con giro de 2,5° y corrección empleando una ventana de 50 ms (frecuencia de muestreo de 1000 Hz.)

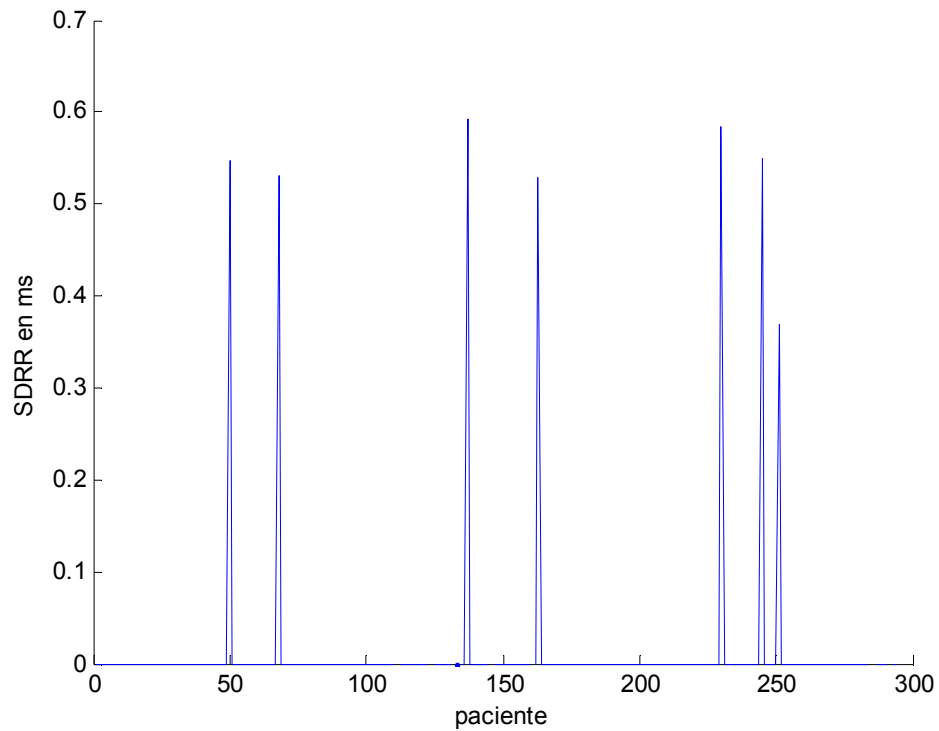


Fig. 80 SDRR en función del paciente para detector Genetic Algorithm con giro de 2,5° y corrección empleando una ventana de 100 ms (frecuencia de muestreo de 1000 Hz.)

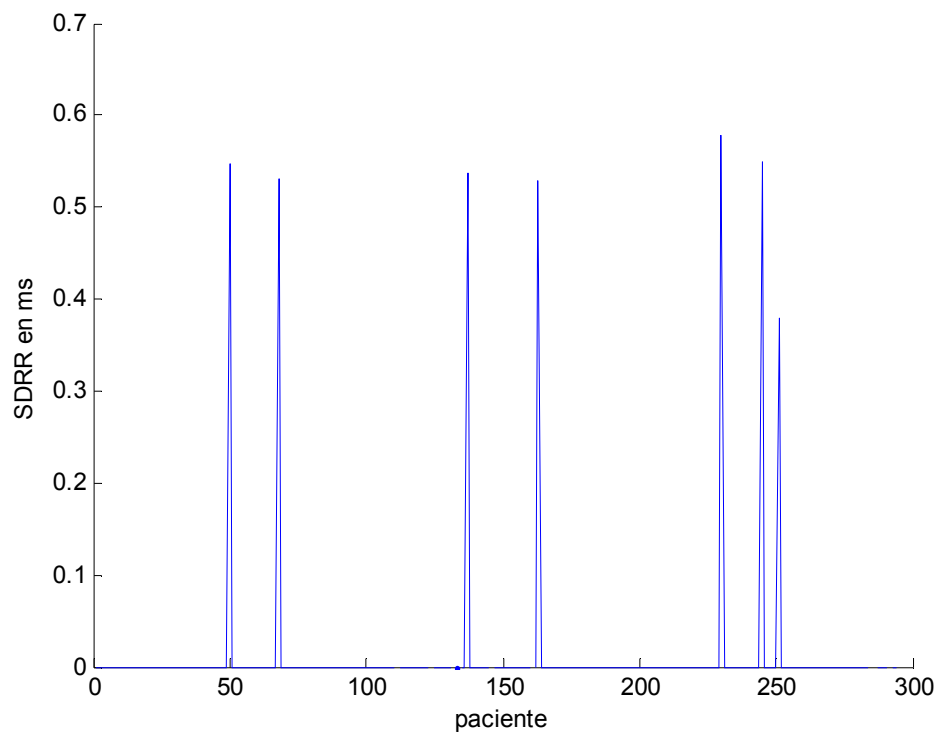


Fig. 81 SDRR en función del paciente para detector Genetic Algorithm con giro de 2,5° y corrección empleando una ventana de 200 ms (frecuencia de muestreo de 1000 Hz.)

Posteriormente, se ha procedido a realizar la misma comparativa para dicho detector pero con giros de $12,5^\circ$.

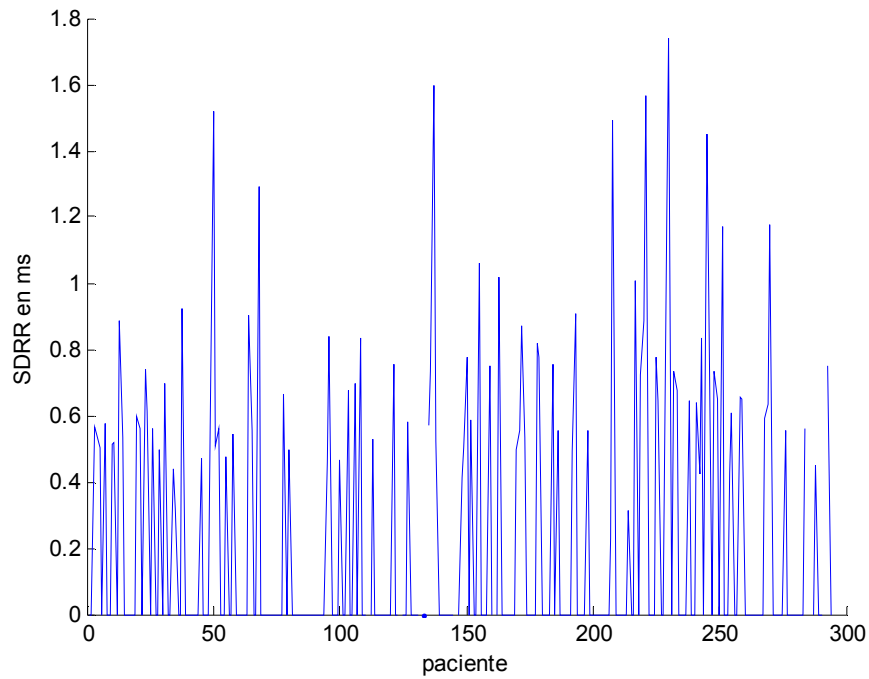


Fig. 82 SDRR en función del paciente para detector Genetic Algorithm con giro de $12,5^\circ$ y corrección empleando una ventana de 50 ms (frecuencia de muestreo de 1000 Hz.)

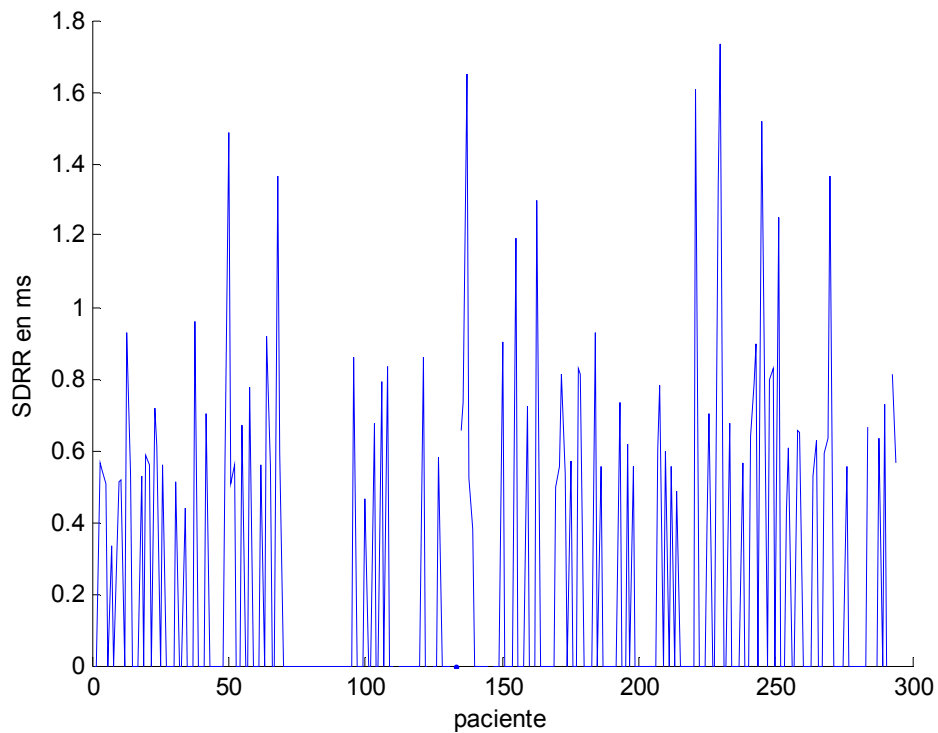


Fig. 83 SDRR en función del paciente para detector Genetic Algorithm con giro de $12,5^\circ$ y corrección empleando una ventana de 100 ms (frecuencia de muestreo de 1000 Hz.)

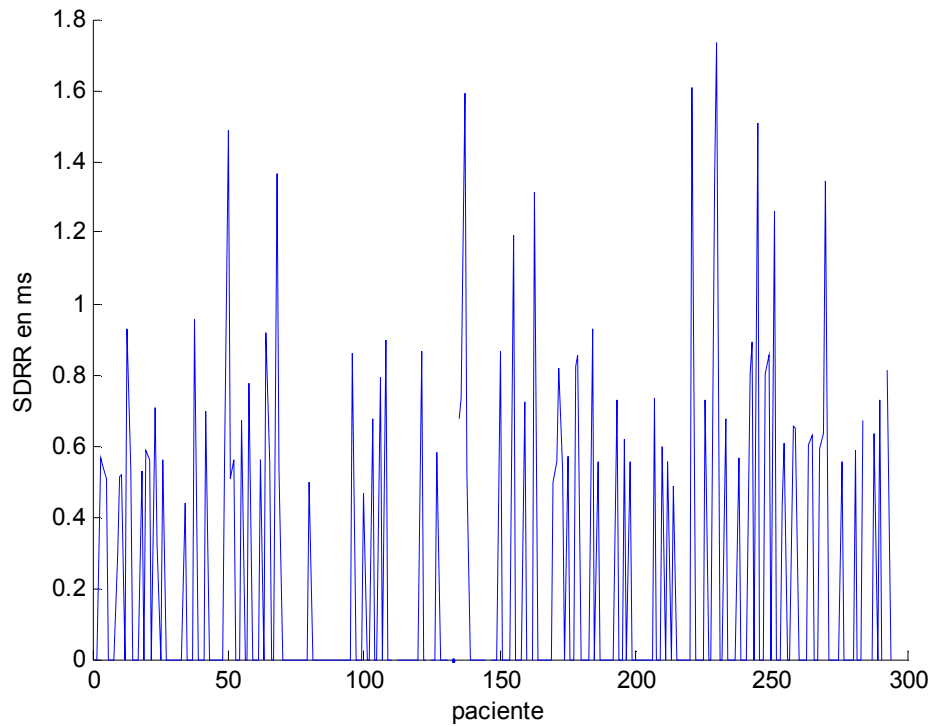


Fig. 84 SDRR en función del paciente para detector Genetic Algorithm con giro de $12,5^\circ$ y corrección empleando una ventana de 200 ms (frecuencia de muestreo de 1000 Hz.)

A continuación, se ha llevado a cabo la misma tarea con el detector de Ruha. En primer término, se ha probado con una ventana de $2,5^\circ$.

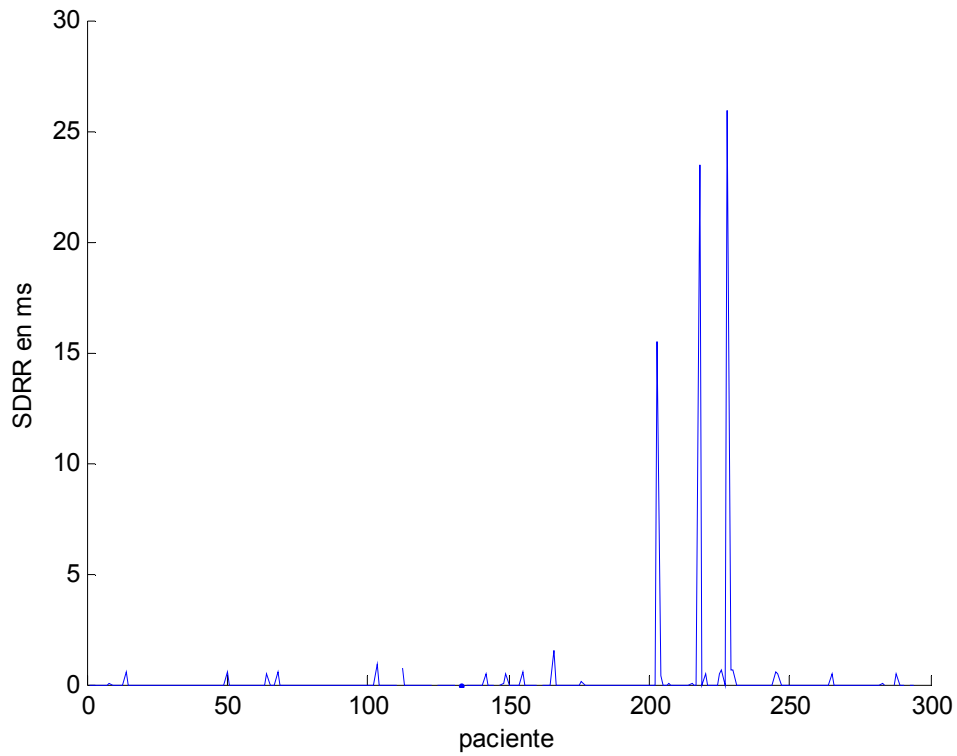


Fig. 85 SDRR en función del paciente para detector Ruha con giro de 2,5° y corrección empleando una ventana de 50 ms (frecuencia de muestreo de 1000 Hz.)

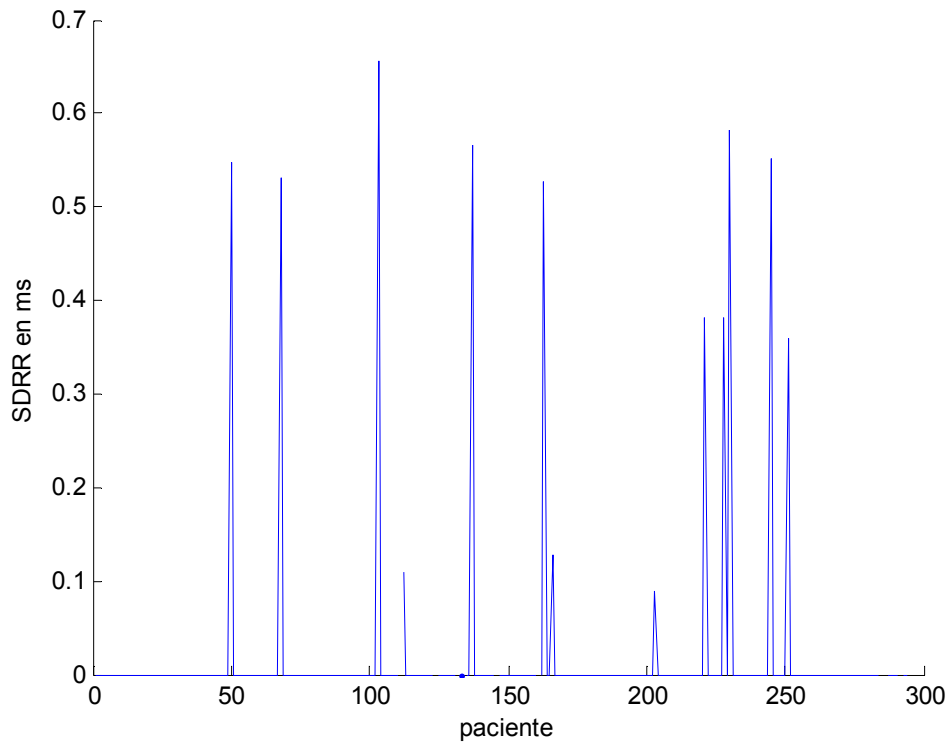


Fig. 86 SDRR en función del paciente para detector Ruha con giro de 2,5° y corrección empleando una ventana de 100 ms (frecuencia de muestreo de 1000 Hz.)

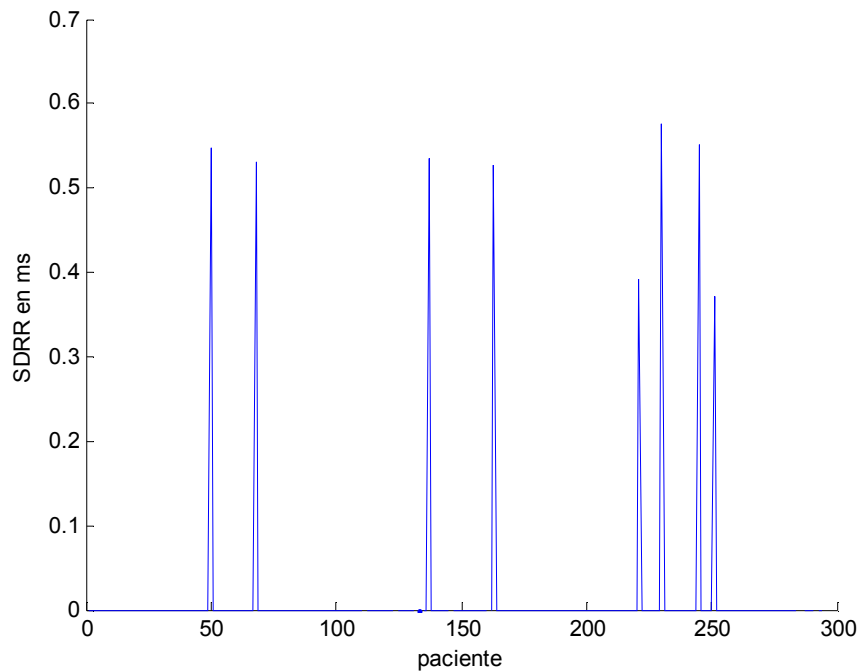


Fig. 87 SDRR en función del paciente para detector Ruha con giro de $2,5^\circ$ y corrección empleando una ventana de 200 ms (frecuencia de muestreo de 1000 Hz.)

Posteriormente se ha probado con un giro de $12,5^\circ$ para el mismo detector.

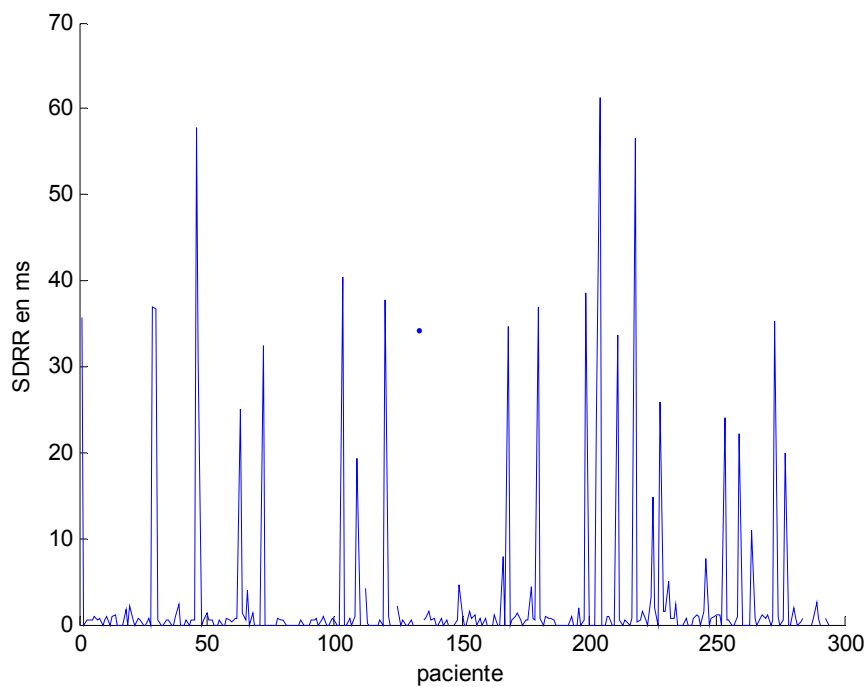


Fig. 88 SDRR en función del paciente para detector Ruha con giro de $12,5^\circ$ y corrección empleando una ventana de 50 ms (frecuencia de muestreo de 1000 Hz.)

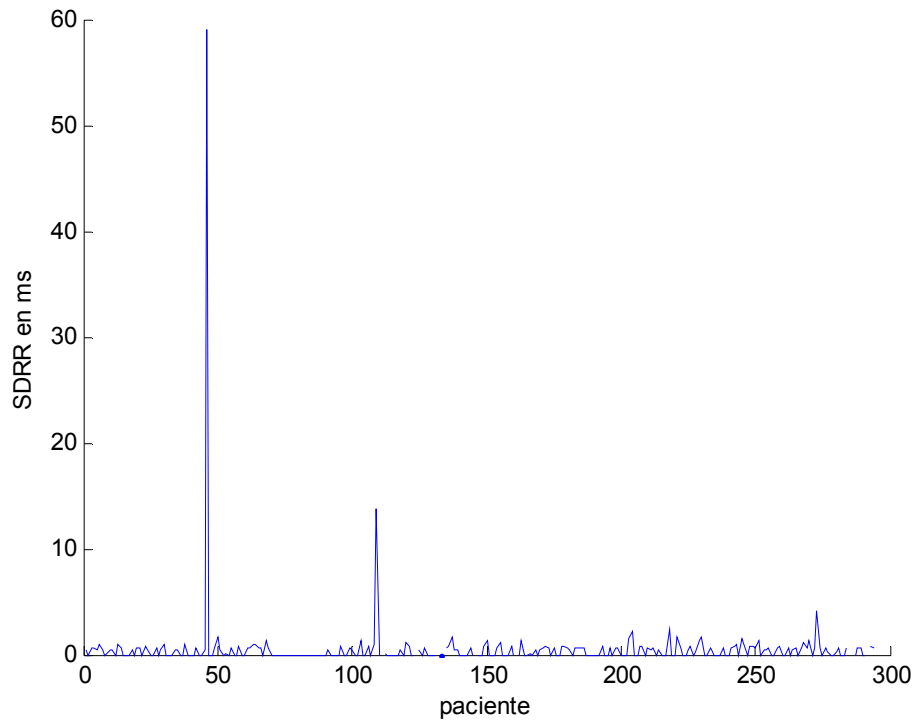


Fig. 89 SDRR en función del paciente para detector Ruha con giro de $12,5^\circ$ y corrección empleando una ventana de 100 ms (frecuencia de muestreo de 1000 Hz.)

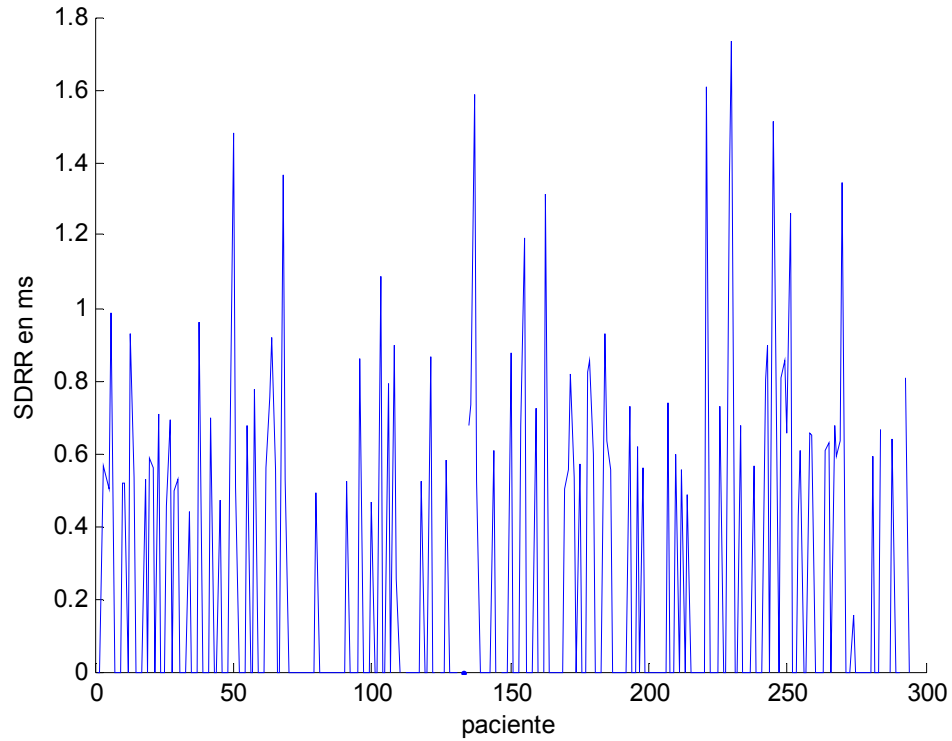


Fig. 90 SDRR en función del paciente para detector Ruha con giro de $12,5^\circ$ y corrección empleando una ventana de 200 ms (frecuencia de muestreo de 1000 Hz.)

Como era de esperar, la variación del parámetro de la ventana con el detector de Ruha no afecta de la misma manera que con el detector Genetic Algorithm. Para el primero, en giros de $2,5^\circ$, cuando se pasa de una ventana de 50 ms a una de 100 ms, se experimenta una disminución de la desviación estándar de algunos pacientes. Sin embargo, cuando se pasa de una ventana de 100 ms a una de 200 ms, los resultados siguen mejorando. Esto se debe a que la magnitud de los errores del detector de Ruha es bastante mayor que el Genetic Algorithm (recordemos que se había escogido el detector Ruha para dicho estudio por su condición de ser el detector más sensible a los cambios de morfología del ECG). Por el otro lado, con el detector Genetic Algorithm y para giros de $2,5^\circ$ se ha observado que con una ventana de 50 ms ya basta. Al incrementar la ventana en muchas ocasiones lo único que sucederá es que se incrementará el error de algunos pacientes en concreto. Asimismo, si se incrementa la magnitud del giro a $12,5^\circ$, el error en la corrección para una ventana de 100 ms es ligeramente inferior para ciertos pacientes. En líneas generales, podemos considerar que con una ventana de 50 ms los resultados obtenidos serán más óptimos.

Con todo, empleando las ventanas óptimas en la corrección obtendremos en la comparativa entre resultados corregidos y resultados sin la corrección lo siguiente (en rojo los resultados sin corregir y en azul las detecciones corregidas):

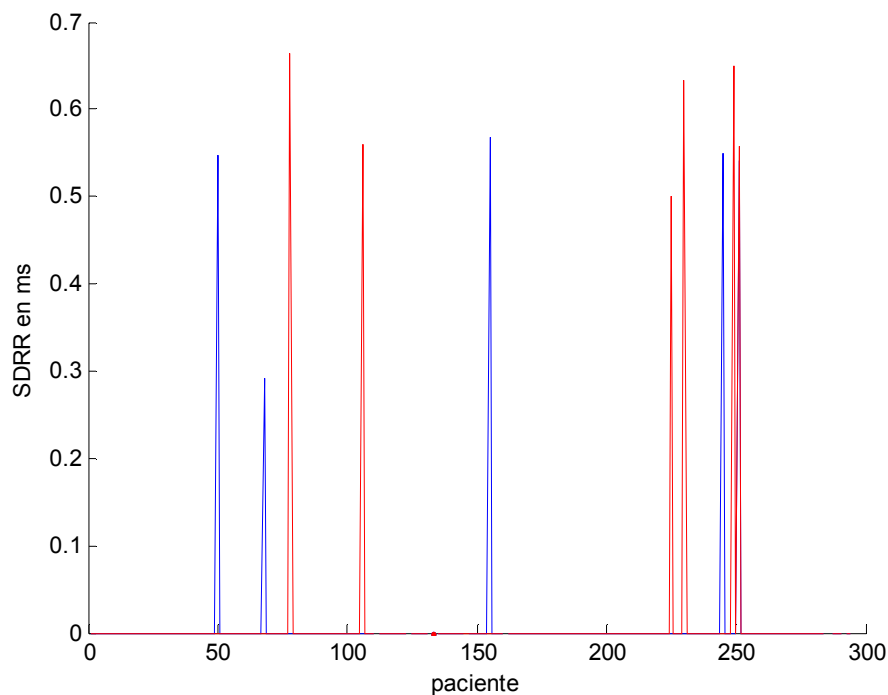


Fig. 91 Comparativa de detección con Genetic Algorithm sin corregir (rojo) y detección con Genetic Algorithm corregido con ventana de 50 ms ($f_m=1000$ Hz) (azul) para giro de $2,5^\circ$.

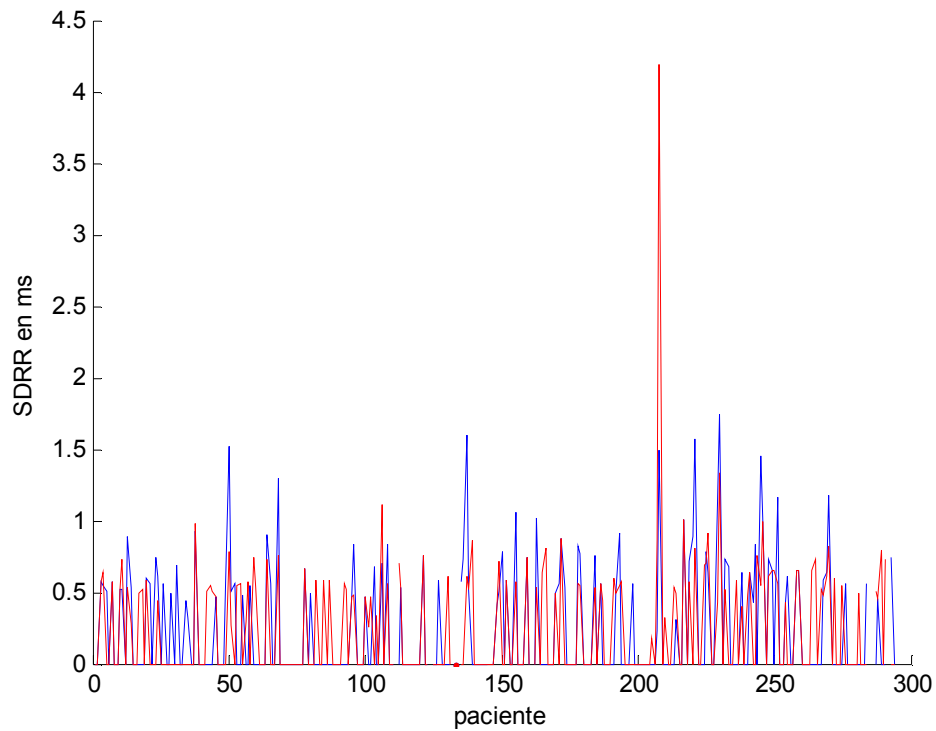


Fig. 92 Comparativa de detección con Genetic Algorithm sin corregir (rojo) y detección con Genetic Algorithm corregido con ventana de 50 ms ($f_m=1000$ Hz) (azul) para giro de $12,5^\circ$.

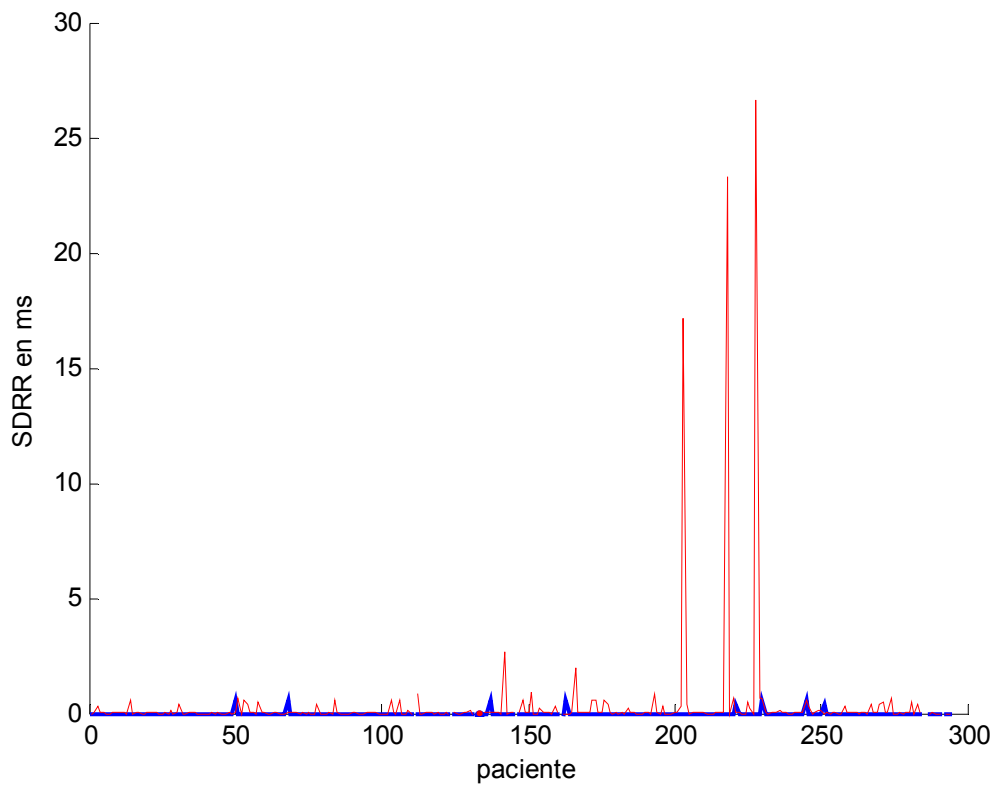


Fig. 93 Comparativa de detección con Ruha sin corregir (rojo) y detección con Ruha corregido con ventana de 200 ms ($f_m=1000$ Hz) (azul) para giro de $2,5^\circ$.

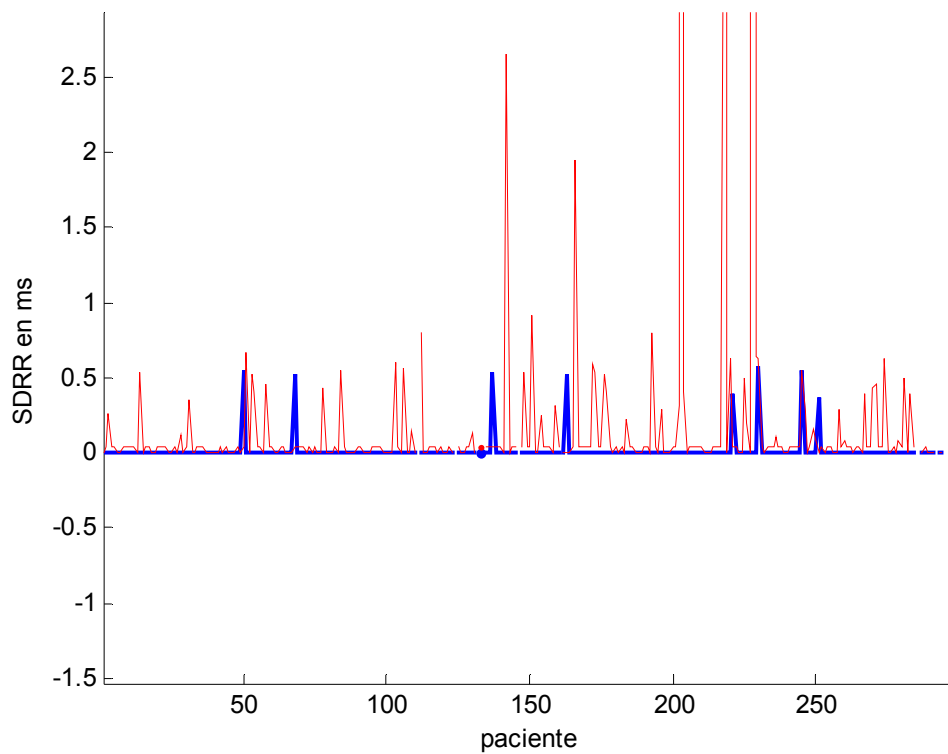


Fig. 94 Zoom de la comparativa de detección con Ruha sin corregir (rojo) y detección con Ruha corregido con ventana de 200 ms ($f_m=1000$ Hz) (azul) para giro de $2,5^\circ$.

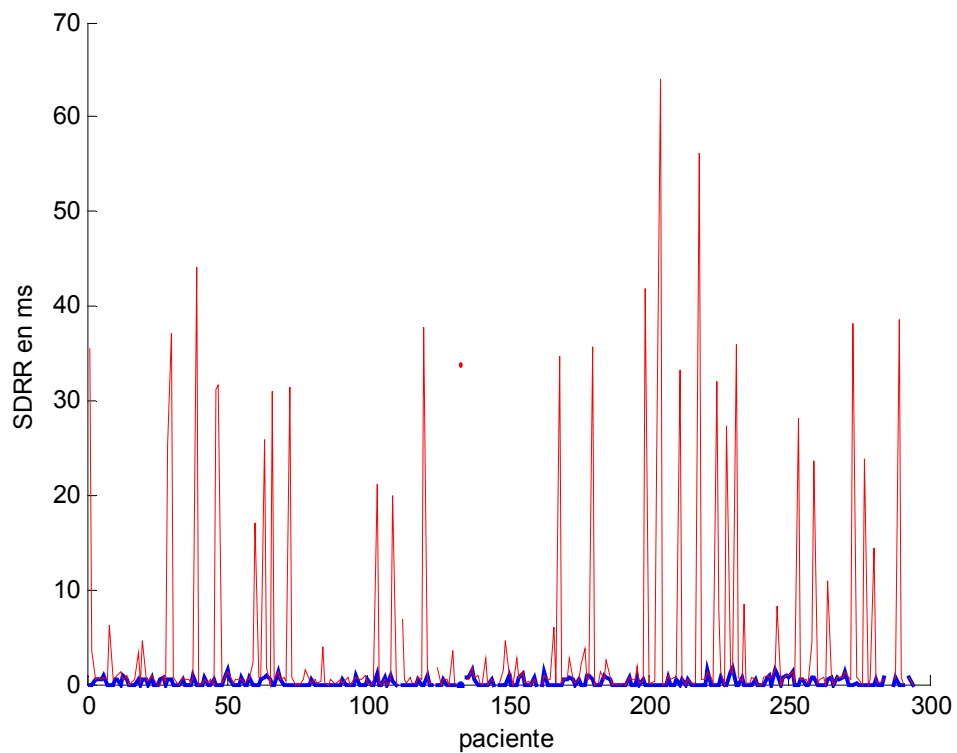


Fig. 95 Comparativa de detección con Ruha sin corregir (rojo) y detección con Ruha corregido con ventana de 200 ms ($f_m=1000$ Hz) (azul) para giro de $12,5^\circ$.

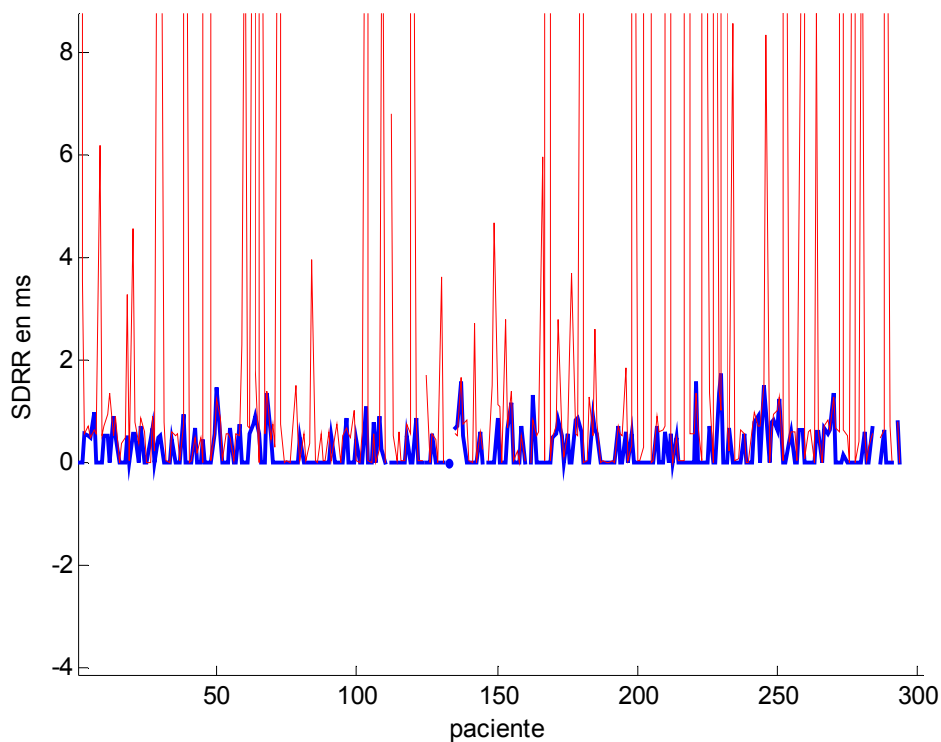


Fig. 96 Zoom de la comparativa de detección con Ruha sin corregir (rojo) y detección con Ruha corregido con ventana de 200 ms ($f_m=1000$ Hz) (azul) para giro de $12,5^\circ$.

Se ha rehecho las tablas pero tomando los resultados para la ventana óptima en función del detector empleado y del giro aplicado. En el caso de Genetic Algorithm se ha obtenido lo siguiente:

	giro $2,5^\circ$	giro $12,5^\circ$
media	0.0088	0.2364
máximo	0.5674	1.7419
mínimo	0	0

	giro $2,5^\circ$	giro $12,5^\circ$
media	0.0125	0.2393
máximo	0.6645	4.1852
mínimo	0	0

Tabla 11. SDRR (ms) media, máxima y mínima para el detector Genetic Algorithm con corrección con ventana óptima (50 ms) comparado con los valores obtenidos para el mismo detector pero sin corrección (tabla en rojo).

En el caso del detector de Ruha tenemos:

	giro 2,5°	giro 12,5°
media	0.0141	0.2561
máximo	0.5756	1.7338
mínimo	0	0

	giro 2,5°	giro 12,5°
media	0.3389	4.1935
máximo	26.6266	64.0639
mínimo	0	0

Tabla 12. SDRR (ms) media, máxima y mínima para el detector Ruha con corrección con ventana óptima (200 ms) comparado con los valores obtenidos para el mismo detector pero sin corrección (tabla en rojo).

A continuación se incluye una comparativa de ambos detectores utilizando sus respectivas ventanas óptimas dado un mismo giro.

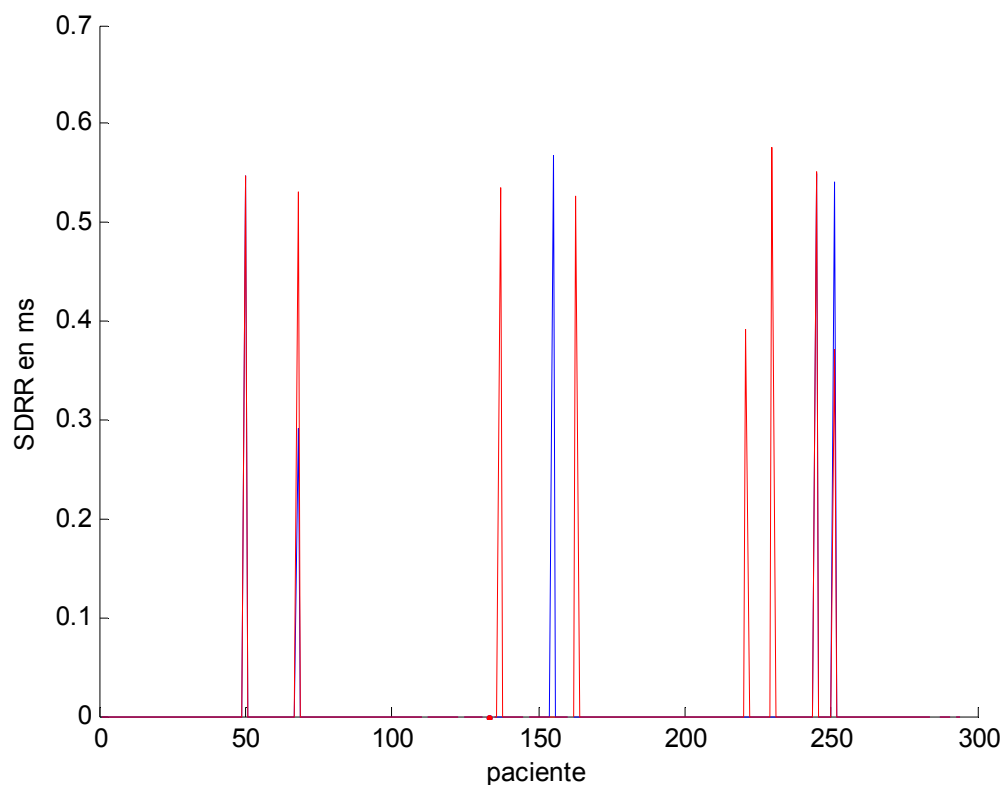


Fig. 97 Comparativa de Genetic Algorithm corregido con ventana de 50 ms (en azul) y Ruha corregido con ventana de 200 ms (en rojo), ambos para giros de 2,5°.

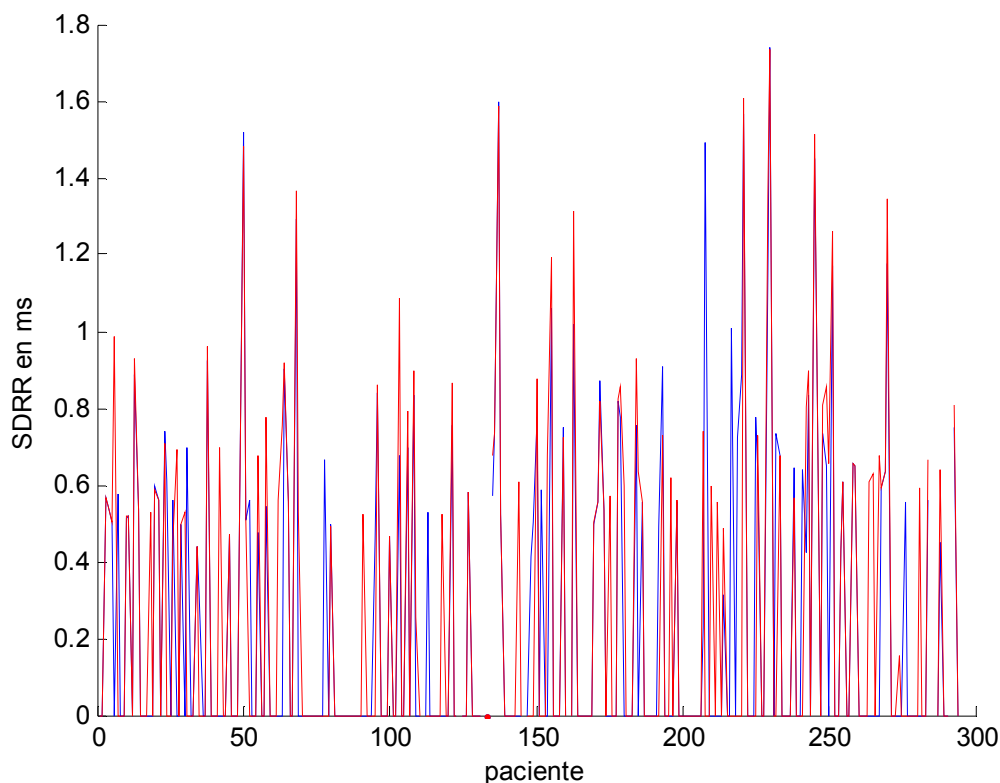


Fig. 98 Comparativa de Genetic Algorithm corregido con ventana de 50 ms (en azul) y Ruha corregido con ventana de 200 ms (en rojo), ambos para giros de 12,5°.

De esta comparativa podemos extraer que tras la corrección por correlación, los errores se han reducido a errores de una misma magnitud para ambos detectores ya que son los debidos a la propia corrección.

Como conclusión de este apartado, tenemos que subrayar que se dan dos casuísticas en cuanto a la corrección por correlación.

Por un lado, tenemos el caso del detector robusto cuyos errores debidos exclusivamente a cambios de morfología son próximos a cero (en nuestro caso 0.0125 ms para el giro pequeño y 0.2364 ms para el giro grande). En esta situación, la inclusión de un corrector por correlación no tiene mucho sentido puesto que el mismo corrector añade incertidumbre. Es decir, que para detectores robustos no sólo no se mejorará sustancialmente los resultados sino que probablemente se empeoren ya que se añade incertidumbre. En nuestro estudio, el detector Genetic Algorithm con la corrección mantendrá unos errores de la misma magnitud que sin ella (con una SDRR media de 0.0088 ms para el giro pequeño y otro de 0.2364 ms para el grande). Si bien, cabe destacar que se ha logrado reducir bastante eficientemente el valor máximo de SDRR

para giros grandes pasando de 4,1852 ms a 1,7419 ms. Es decir, que la corrección funciona correctamente para reducir SDRRs de más de 2 ms. El hecho de que los máximos hayan disminuido mientras se mantiene la media de la SDRR nos hace intuir que es posible que otros pacientes hayan visto incrementados sus errores. En la figura 92 se puede apreciar que para errores menores que 2 ms puede ser que se vean corregidos pero también que se vean incrementados.

Asimismo, otra lectura que se puede realizar a la luz de estos resultados es que en el caso de giros de 12,5° la incertidumbre introducida por el corrector es mayor que en el caso de giros de 2,5°. El hecho de correlar con una señal que sufre un cambio de morfología más grande conlleva una mayor incertidumbre del proceso ya que los complejos QRS con los que se correla la máscara son menos parecidos entre sí que en los giros de 2,5°. Es decir, el corrector también es sensible a los cambios de morfología del ECG y, en consecuencia, la corrección será menos eficiente cuanto más patentes sean dichos cambios.

Por el otro lado, está el caso del detector poco robusto, el detector de Ruha. Los resultados obtenidos para este detector son bastante buenos puesto que se rebaja sustancialmente la desviación estándar media de las señales RR. En el caso del giro de 2,5° pasa de 0.3389 ms a 0.0141ms y en el giro de 12,5° pasa de 4.1935 ms a 0,2561 ms. Parece evidente que en el caso de detectores poco robustos, la corrección por correlación sí que contribuye a fortalecer la detección frente a cambios de morfología del ECG hasta tal punto que los resultados son de un mismo orden que los obtenidos por el detector robusto con corrección. Los errores que se aprecian son los que restan de la corrección por correlación que, a su vez, ha minimizado los debidos a una detección poco robusta a cambios de morfología. A pesar de que la incertidumbre introducida para giros grandes sea, como en el caso del detector Genetic Algorithm, mayor que para giros pequeños, ésta es bastante menor que el error ligado a una detección poco robusta frente a cambios de morfología del ECG. En consecuencia, intuimos que una corrección por correlación para detectores poco robustos (tanto para giros pequeños como para giros muy grandes) normalmente mejorará su rendimiento.

Además, la elección de una ventana adecuada a las características del detector y a la magnitud de los giros también será un parámetro que mejorará la detección de los complejos QRS. En general, para detectores menos robustos parece lógica la elección de una ventana mayor (y, consecuentemente, con mayor capacidad de corregir errores de mayor magnitud). En el caso del detector de Ruha ha sido necesaria una ventana

temporal de 200 ms mientras que para el detector Genetic Algorithm con una de 50 ms ya bastaba.

Por último, cabe destacar que se dan casos en los que las características que presenta el propio latido del paciente hacen que el ECG sintético creado sea más sensible a cambios de morfología inducidos. Por ejemplo, en las siguientes dos figuras podemos ver como el cambio de morfología afecta en especial medida. Se trata de la derivación x del paciente 41. Para el giro de $12,5^\circ$, la relación señal a ruido es mucho menor que para el giro de $2,5^\circ$.

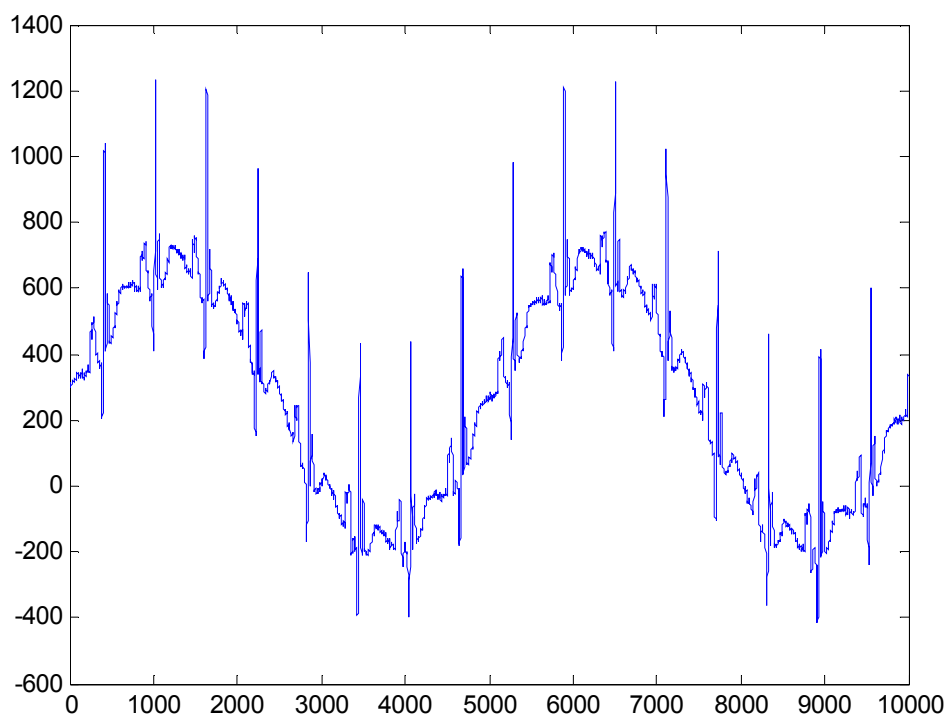


Fig. 99 Derivación x del paciente 41 girado $2,5^\circ$.

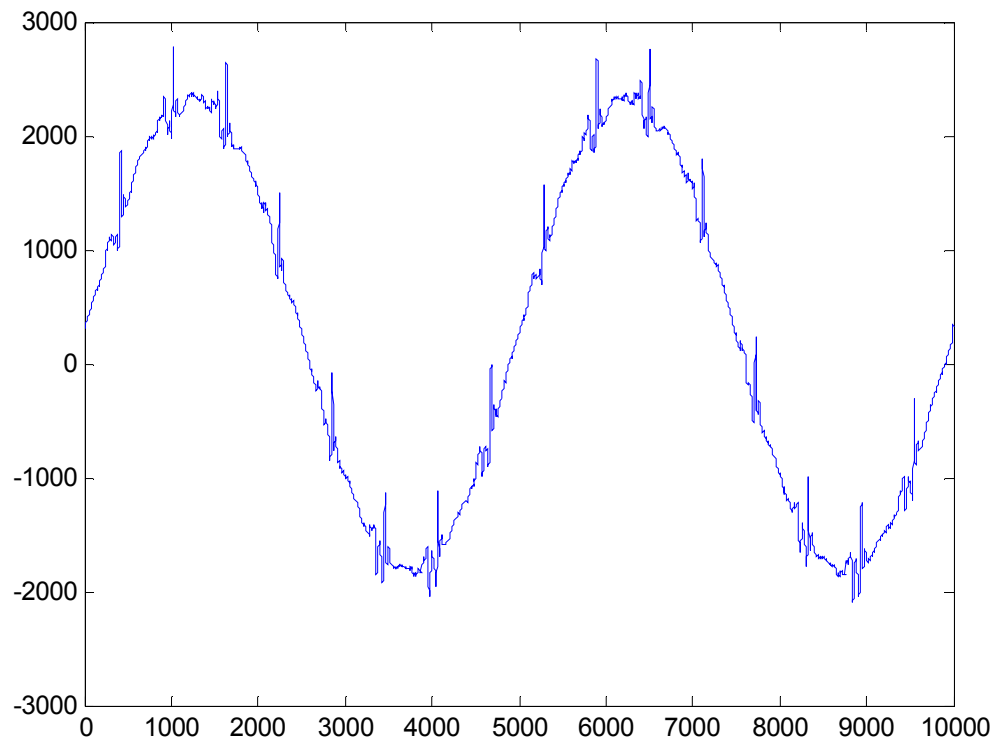


Fig. 100 Derivación x del paciente 41 girado $12,5^\circ$. La relación señal a ruido ha sido muy deteriorada por el incremento en la magnitud de la rotación.

12. Aplicación del proceso de detección de complejos QRS a pacientes transplantados de corazón

A continuación, se ha procedido a la detección de complejos QRS para señales ECG de pacientes que han sido transplantados del corazón. Se llevará a cabo todo el proceso realizado hasta ahora en el estudio. Es decir, una primera detección seguida de una precorrección y, por último, una corrección por medio del método de correlación.

En cuanto a los ECGs pertenecientes a pacientes transplantados del corazón es importante tener en cuenta que se trata de los pacientes que presentan menor variabilidad de ritmo cardíaco. En consecuencia, se tratará de un caso práctico muy particular donde podremos discernir mejor los efectos que tienen los cambios de morfología del ECG sobre la sensibilidad de los detectores. En total hemos cogido los electrocardiogramas de 70 pacientes transplantados.

Previamente a toda realización se ha observado que hay pacientes que padecen arritmias como las del paciente 12, tal y como podemos ver en las siguientes dos figuras:

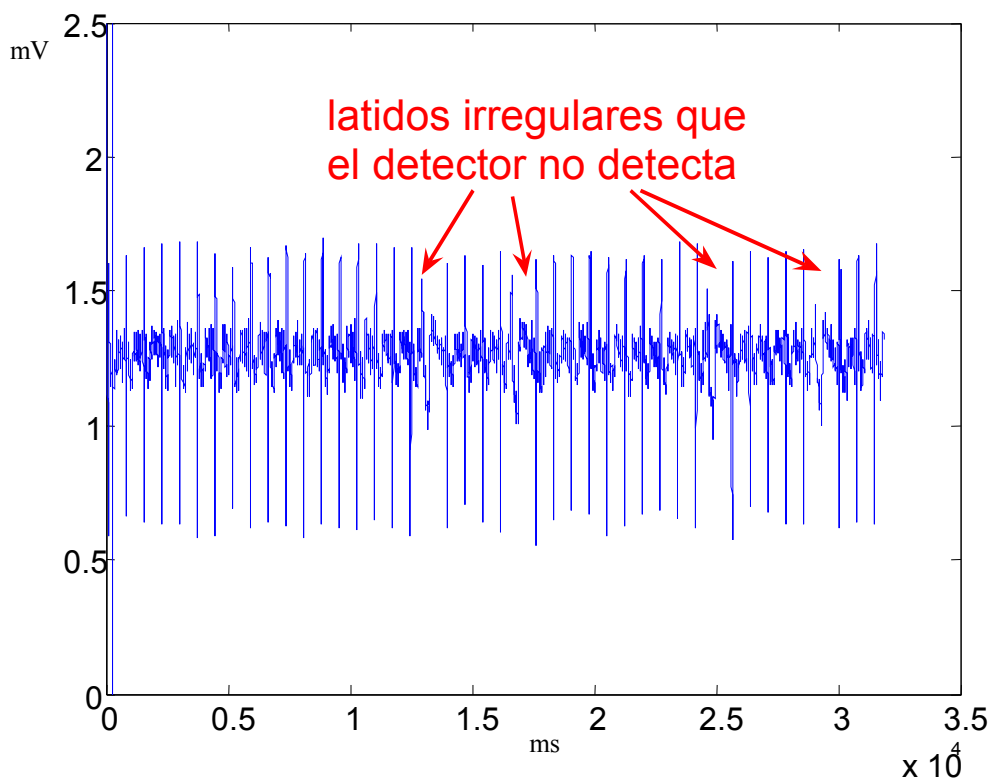


Fig. 101 Señal ECG del paciente 12 en el que se puede apreciar algunos latidos irregulares que condicionan la detección y su posterior valoración en la SDRR.

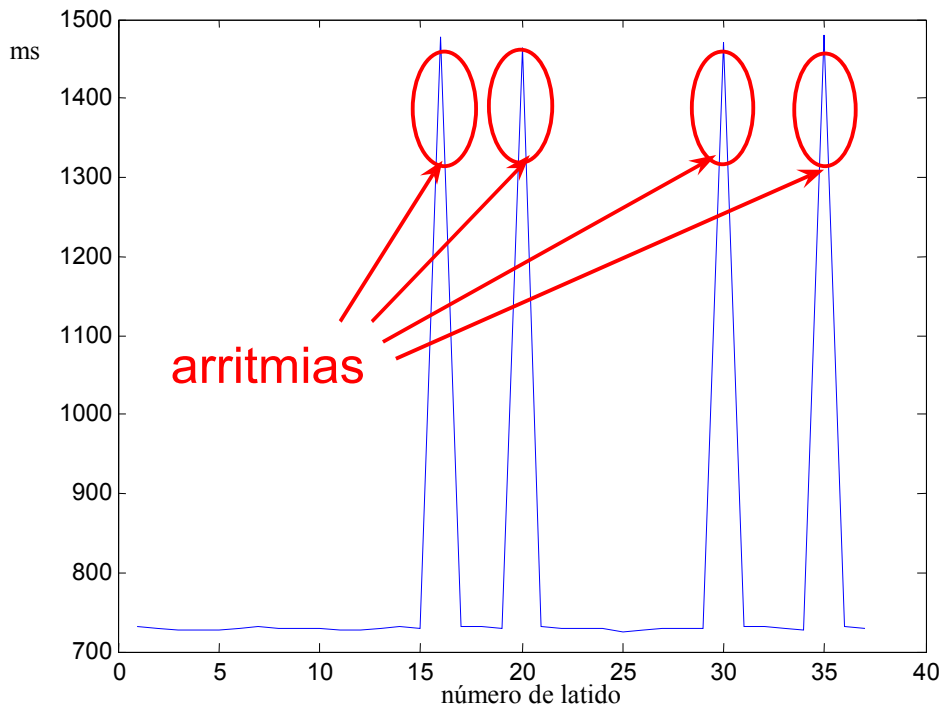


Fig. 102 Serie RR del paciente 12 en el que se puede apreciar como las arritmias afectan a su SDRR solapando los efectos de los cambios de morfología.

El aumento de la desviación estándar inducido por estas arritmias en la variabilidad del ritmo cardíaco sería de tal magnitud que los errores derivados de los cambios de morfología del ECG quedarían totalmente tapados. Por consiguiente, se ha optado por eliminar dichos casos del estudio. Concretamente, se ha descartado a los pacientes 12, 15, 16, 17, 18 y 43. Asimismo, el paciente número 58 también ha sido obviado debido a que su ECG presentaba una peor relación señal a ruido que producía un error mucho mayor que el resto (y que desvirtuaría los resultados obtenidos).

Cabe apuntar que los cambios de morfología que se pueden apreciar en los pacientes transplantados tienen un perfil más próximo al cambio de morfología de $2,5^\circ$ (magnitud próxima a la causada por la respiración estándar del individuo), inducido artificialmente en los estudios anteriores a la hora de crear ECGs sintéticos, que al de $12,5^\circ$ (que, recordemos, se trataba de un ejemplo exagerado).

La primera fase del estudio consistirá en la evaluación de los detectores frente a los ECGs de dichos pacientes sin emplear el corrector por correlación. Posteriormente, se introducirá dicha corrección empleando una ventana de 50 muestras para el detector Genetic Algorithm y otra de 200 muestras para el detector Ruha (con una frecuencia de muestreo de 1000 Hz). Las gráficas obtenidas son las siguientes:

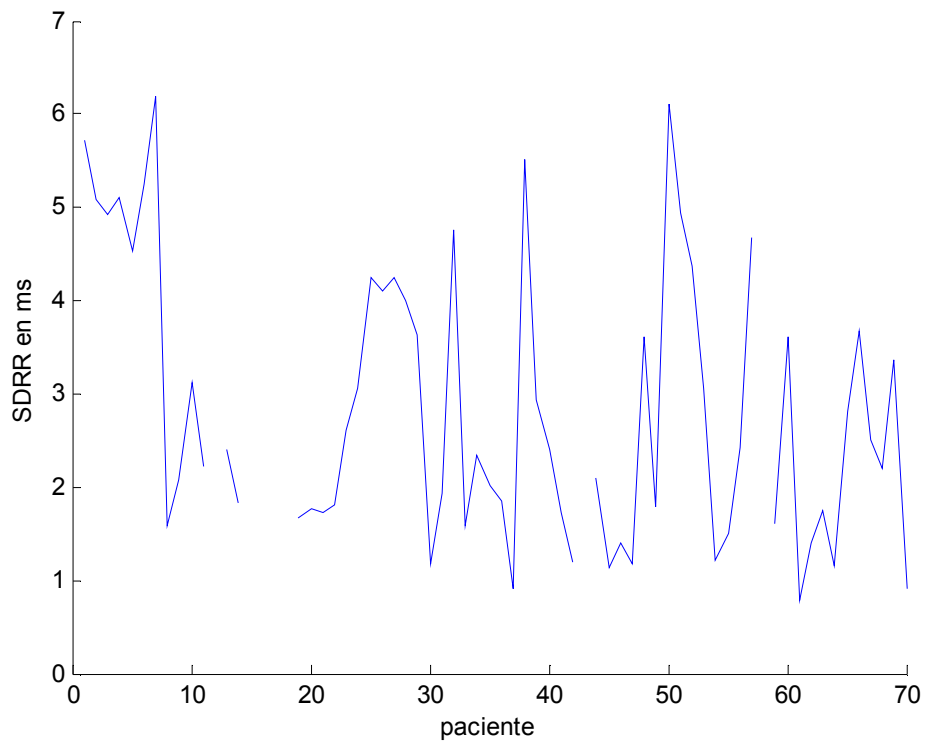


Fig. 103 SDRR para pacientes transplantados tras detección con Genetic Algorithm y sin corrección.

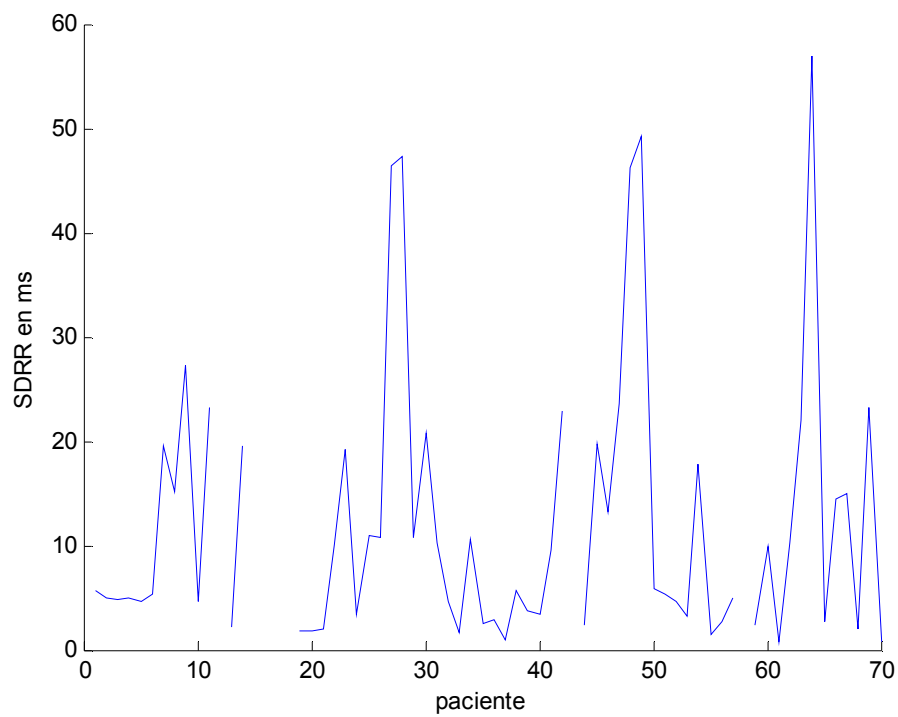


Fig. 104 SDRR para pacientes transplantados tras detección con Ruha y sin corrección.

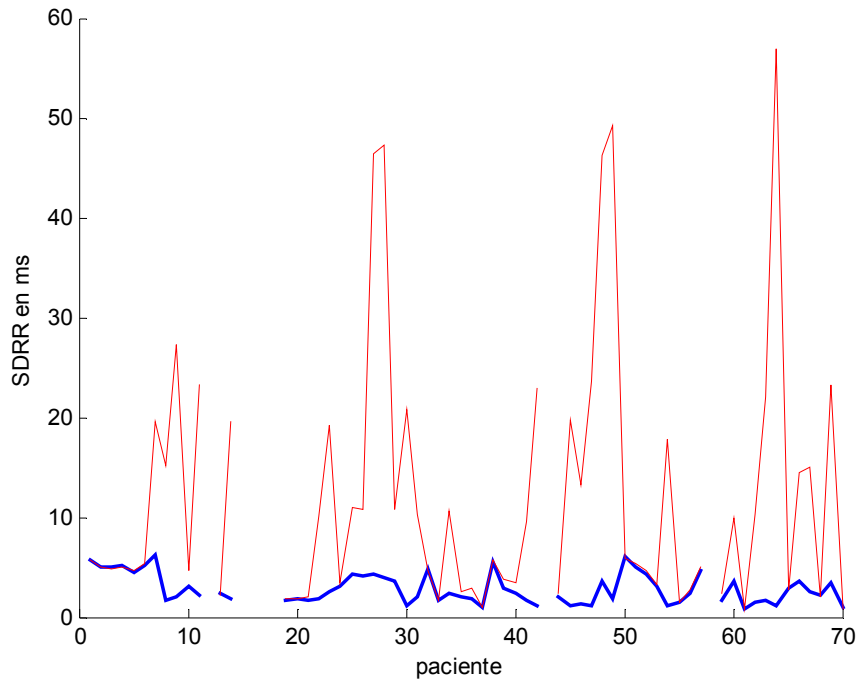


Fig. 105 Comparativa de resultados de Genetic Algorithm sin corregir (azul) con Ruha sin corregir (rojo).

	Genetic Algorithm	Ruha
media	2.8358	12.1952
máximo	6.1855	56.8822
mínimo	0.7782	0.7582

Tabla 12. SDRR (ms) media, máxima y mínima para detectores Genetic Algorithm y Ruha sin corrección.

En esta última gráfica se puede apreciar como las desviaciones estándar de la señal RR para el detector Genetic Algorithm son, en general, bastante menores que para el detector Ruha. Dado que los pacientes transplantados de corazón son los que tienen menor variabilidad del ritmo cardíaco, intuimos que dicho aumento es debido a la distinta robustez frente a cambios de morfología del ECG que tienen los detectores empleados. Como era de esperar, el detector que se ha mostrado menos robusto en el caso teórico (Ruha) produce en el caso práctico más error que el detector robusto (Genetic Algorithm).

Si los resultados entre ambos detectores (sin llevar a cabo el proceso de corrección) hubieran sido comparables querría decir que los cambios en la morfología del ECG no afectan determinantemente en pacientes transplantados y, en consecuencia, el error producido hubiera podido ser considerado negligible en las investigaciones.

El siguiente paso ha sido el de corregir las detecciones por medio del corrector por correlación y comparar dichos resultados con los obtenidos sin corregir. Para el detector Genetic Algorithm se ha obtenido lo siguiente:

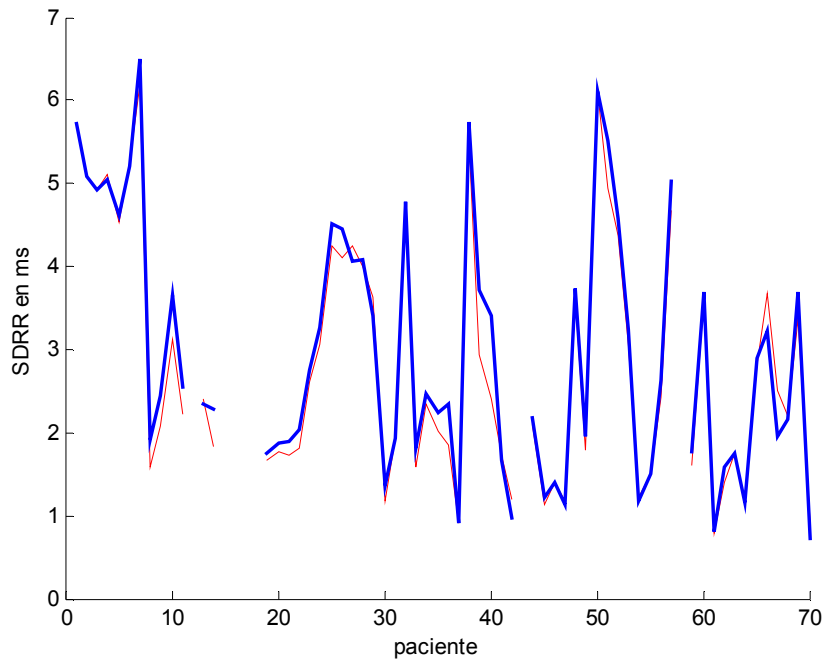


Fig. 106 Comparativa Genetic Algorithm corregido (azul) y sin corregir (rojo).

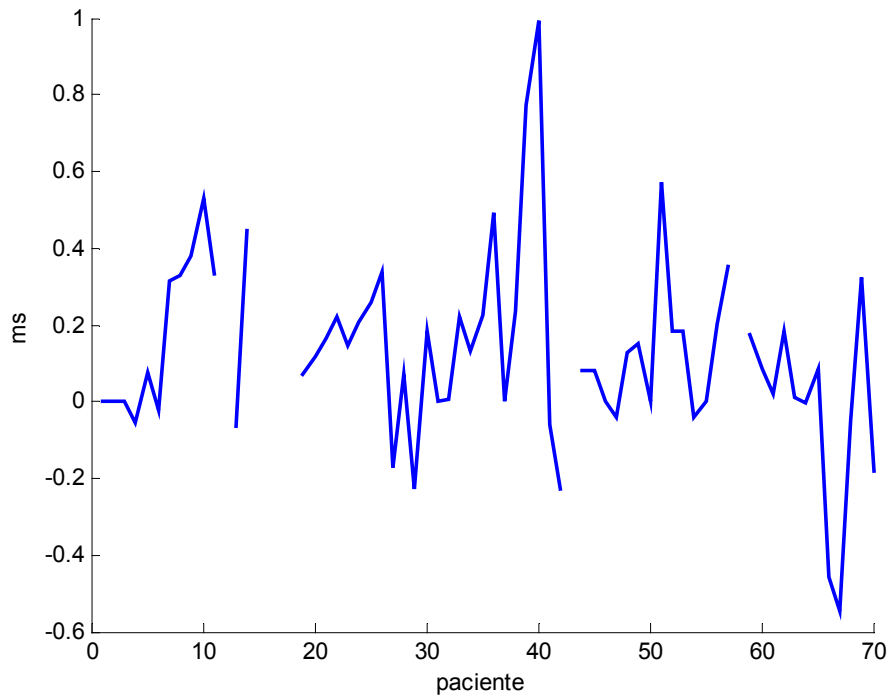


Fig. 107 Gráfica de la diferencia entre los resultados para Genetic Algorithm corregido y sin corregir.

	sin corregir	corregido
media	2.8358	2.9616
máximo	6.1855	6.4988
mínimo	0.7782	0.7314

Tabla 13. SDRR (ms) media, máxima y mínima para el detector Genetic Algorithm con corrección y sin corrección.

Análogamente, se ha llevado a cabo el mismo procedimiento para el detector de Ruha resultando lo que sigue:

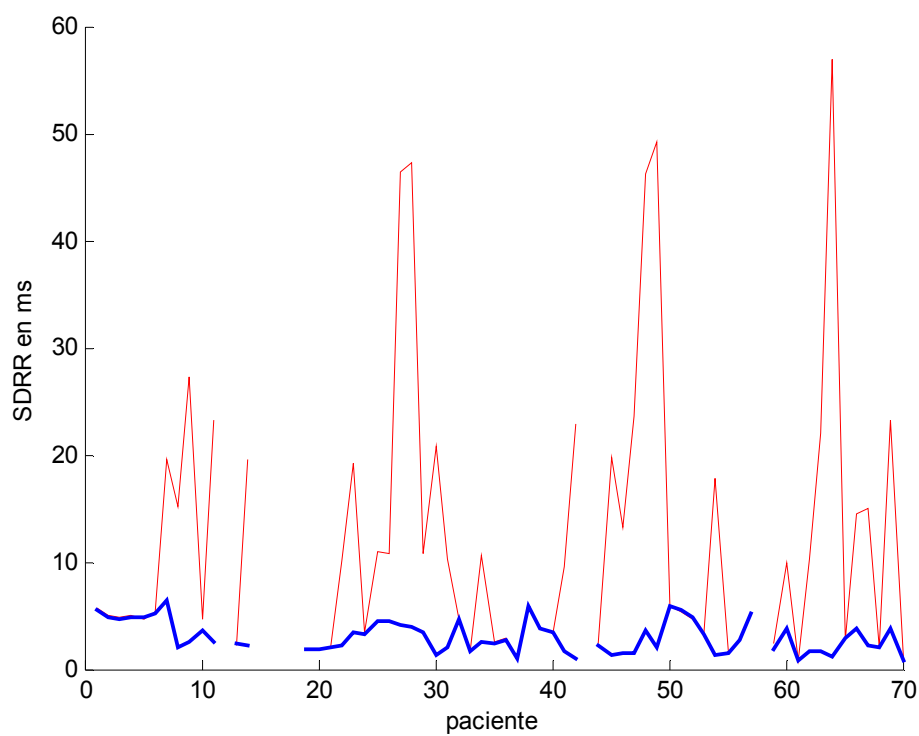


Fig. 108 Comparativa Ruha corregido (azul) y sin corregir (rojo).

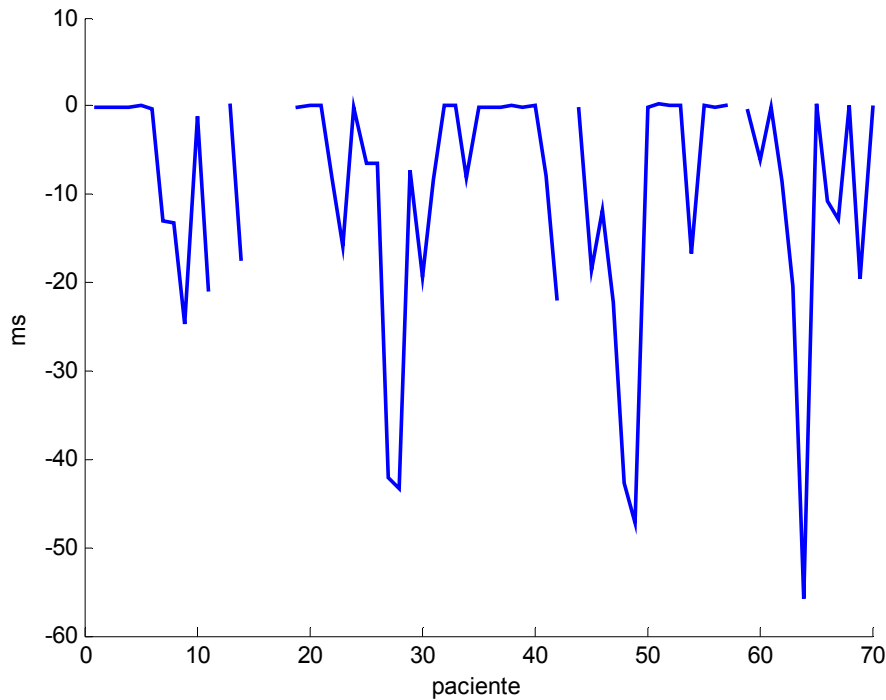


Fig. 109 Gráfica de la diferencia entre los resultados para Ruha corregido y sin corregir.

	sin corregir	corregido
media	12.1952	2.9946
máximo	56.8822	6.4380
mínimo	0.7582	0.7035

Tabla 14. SDRR (ms) media, máxima y mínima para el detector Ruha con corrección y sin corrección.

En el caso del detector Genetic Algorithm, la media de diferencia de la corrección es de 0.1258 ms. Esto quiere decir que, en general, la SDRR tras la corrección será ligeramente mayor que la SDRR sin corregir. A pesar de que se den casos en los que la corrección mejore ligeramente los resultados, parece lógico pensar que normalmente será contraproducente llevar a cabo una corrección para detectores robustos puesto que los resultados difícilmente mejorarán sustancialmente. Dicho resultado concuerda con el obtenido en el apartado anterior para detectores robustos con ECGs sintéticos sometidos a giros pequeños.

En el otro extremo, encontramos el detector Ruha cuyos resultados sí experimentan una mejora sustancial tras la corrección. La media de la diferencia entre los resultados corregidos y los no corregidos es de -9.2006 ms (es decir, en promedio el corrector reducirá en 9.2006 ms la desviación estándar de la señal RR del paciente). El

hecho de que se trate de un detector bastante sensible a cambios de morfología del ECG hace que los resultados en la detección sean muy susceptibles a ser mejorados. En consecuencia, será conveniente incluir una fase de corrección tras la detección en el caso de detectores poco robustos.

Por último, se ha confrontado los resultados de ambos detectores una vez corregidos por correlación. Se ha obtenido la siguiente gráfica:

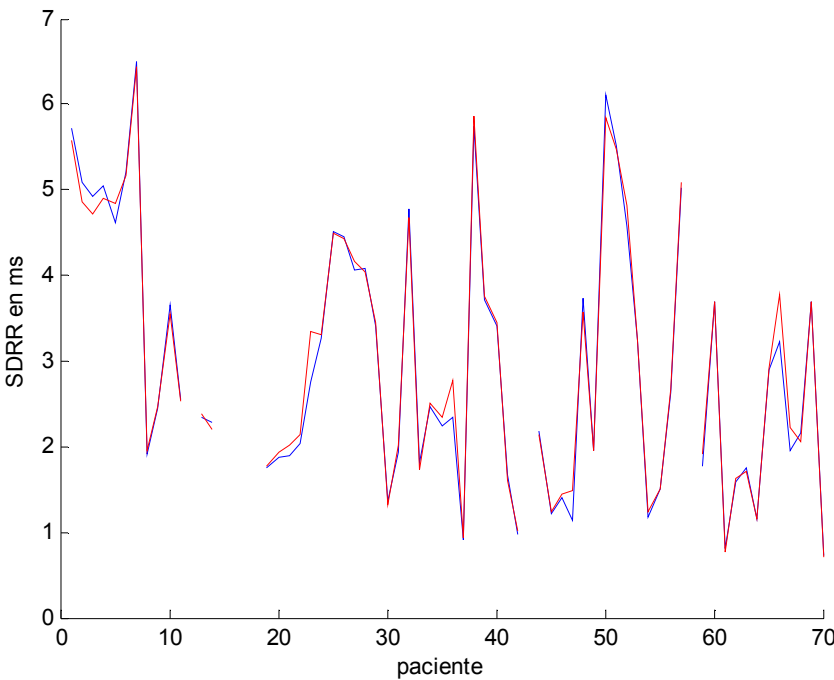


Fig. 110 Comparativa de los resultados de Genetic Algorithm corregido (azul) y Ruha corregido (rojo).

	Genetic Algorithm	Ruha
media	2.9616	2.9946
máximo	6.4988	6.4380
mínimo	0.7314	0.7035

Tabla 15. SDRR (ms) media, máxima y mínima para detectores Genetic Algorithm y Ruha con corrección.

Los resultados de ambos detectores tras la corrección por correlación son muy similares. Aunque, en media, son ligeramente mejores los del detector Genetic Algorithm, ambos presentan errores de un mismo orden. En consecuencia, podemos concluir que el error debido a cambios de morfología ha sido minimizado hasta tal punto que apenas sí es perceptible. Así pues, la introducción de una corrección por

correlación ha propiciado que los efectos de cambios de morfología del ECG (que se manifestaban sobre todo en el detector Ruha) puedan ser considerados negligibles. Los errores que muestran la tabla 15 se achacan, pues, a la mínima variabilidad del ritmo cardíaco que presentan los pacientes transplantados (que a pesar de ser los que presentan menos siempre será distinta a cero, el caso ideal).

Cabe destacar que los resultados obtenidos en dicho caso práctico son coherentes con los del apartado anterior para el caso ideal de ECGs sintéticos. El comportamiento de ambos detectores es muy parecido en los dos contextos.

Por último, si comparamos los resultados de nuestro ejercicio práctico con los del apartado anterior donde se trataba un caso ideal obtenemos lo siguiente:

	Genetic Algorithm	Ruha
media	2.9616	2.9946
máximo	6.4988	6.4380
mínimo	0.7314	0.7035

(Giro 2,5°)	Genetic Algorithm	Ruha
media	0.0088	0.0141
máximo	0.5674	0.5756
mínimo	0	0

Tabla 16. Comparativa de los SDRR (ms) media, máxima y mínima para detectores Genetic Algorithm y Ruha con corrección en el caso real de pacientes transplantados (tabla en negro) y en el caso ideal de ECGs sintéticos con cambios de morfología introducidos artificialmente (tabla en rojo).

Como se puede ver, tanto los errores medios obtenidos para Genetic Algorithm como para Ruha en el caso de los ECGs sintéticos (recordemos que se trata del caso ideal en el que la variabilidad del ritmo cardíaco es nula) del apartado anterior son de dos órdenes de magnitud inferiores a los adquiridos en el caso práctico de pacientes transplantados. Dado que éstos últimos se consideran como unos de los que presentan menor variabilidad de ritmo cardíaco, podemos concluir que si se trata de un detector robusto frente a cambios de morfología del ECG los errores debidos a pequeños cambios de morfología (como podrían ser los debidos a la propia respiración del sujeto) no tendrán un peso importante en los resultados finales frente a los errores derivados de la variabilidad del ritmo cardíaco presentado por el paciente. Asimismo, si se trata de un detector sensible a los cambios de morfología del ECG, incluyendo una fase de corrección tras la detección se podrá despreciar el error debido a dichos cambios.

13. Conclusiones

EL ESTUDIO realizado nos ha permitido comprender mejor los distintos grados de sensibilidad que presentan los detectores frente a cambios de morfología del ECG que, a su vez, determinarán la necesidad o no de una fase de corrección.

EN LA PRIMERA FASE del trabajo se ha generado los detectores y se ha comprobado su correcto funcionamiento para sujetos cuya buena salud ha sido contrastada. Se ha obtenido unos parámetros de sensibilidad y de predictibilidad positiva superiores al 99% en la detección de complejos QRS.

Posteriormente, se ha procedido a generar un espacio de experimentación teórico en el que la variabilidad de ritmo cardíaco de los electrocardiogramas fuese nula y donde se pudiera introducir cambios de la morfología del ECG de magnitud graduable. La generación de ECGs sintéticos nos ha permitido verificar los efectos estrictamente relacionados con cambios de morfología del ECG. Se ha usado una base de datos compuesta por pacientes que han padecido distintas patologías del corazón y cuyo ECG en forma de vectorcardiograma permitía la introducción de cambios de morfología a través de una rotación en cada uno de sus ejes.

En la primera exploración sobre los ECGs sintéticos de los diez primeros pacientes se ha llegado a un par de conclusiones. Por un lado, los cambios de morfología del ECG afectan de forma distinta a cada detector. Es decir, hay algunos detectores que son bastante robustos a éstos mientras que otros se muestran muy sensibles. Generalmente, un aumento en la magnitud de los cambios de morfología se verá reflejado en un aumento de la SDRR en la detección (un aumento en mayor o menor medida en función de la robustez del detector). Por el otro lado, el deterioro de la relación señal a ruido en los ECG no será la misma para cada paciente (habrá algunos casos más conflictivos que otros).

De este primer ejercicio se ha podido apreciar que el detector Genetic Algorithm era el más robusto mientras que el detector Ruha figuraba como el más sensible. Al mismo tiempo, otros detectores que se han mostrado fiables son el Wavelet, Hilbert, Zero Crossing, Energy y Hamilton. El detector Length presentaba unos resultados a medio camino entre los anteriormente citados y Ruha. Trahanias ha sido excluido del estudio puesto que se ha considerado sus resultados como aberrantes.

A continuación, se ha usado solamente el detector de mayor y menor robustez con el objeto de introducir un mayor contraste en el estudio a la vez que se limitaba la casuística. Al aumentar el espacio muestral a un total de 294 pacientes se ha podido corroborar la coherencia de los resultados con los ya obtenidos para diez pacientes.

ULTERIORMENTE, se ha valorado la necesidad de introducir una etapa de corrección de los incrementos de la SDRR debidos a los cambios de morfología del ECG. En primer lugar, se ha verificado que el tamaño de la ventana del corrector deberá ser proporcional a la magnitud de los errores a corregir (que, a su vez, dependen tanto de la robustez de los detectores como de la magnitud de los cambios de morfología presentes en el ECG). La corrección será eficiente siempre y cuando se ajuste dicho parámetro a un valor óptimo dada una casuística concreta.

Con todo, los resultados obtenidos reflejan que el corrector por correlación también es sensible a los cambios de morfología. O sea, el corrector se muestra más eficiente con cambios de morfología pequeños que con cambios de morfología grandes ya que en éstos últimos los complejos QRS sufren una mayor deformación.

Asimismo, la necesidad de una etapa de corrección dependerá tanto de la robustez del detector frente a cambios de morfología del ECG como de la magnitud de éstos. En nuestro caso, los resultados obtenidos por el detector robusto no se veían sustancialmente mejorados tras la detección. Incluso en algunos casos se veía empeorado puesto que el propio corrector es una fuente de error. Si se trata de un detector poco robusto, la corrección será más conveniente cuanto mayor sea el giro aplicado. Por consiguiente, si la magnitud del error introducido por el corrector por correlación es comparable o mayor al error del detector, entonces se puede afirmar que su inclusión será prescindible. Si, por el contrario, el error introducido en la corrección es bastante menor que el error en detección debido a cambios de morfología del ECG, entonces sí que será eficiente poner una etapa de corrección tras la detección.

Además, tras la corrección se equiparan en cuanto a magnitud del error los resultados para el detector robusto y el sensible. Es decir, que la SDRR debida a cambios de morfología apenas sí será perceptible y no tendrá que ser tomada en cuenta en posteriores estudios si la magnitud de lo que se desee medir provoque una SDRR mucho mayor. Quedará como error residual el del propio corrector debido a los cambios de morfología.

EL ÚLTIMO ESTADIO de nuestro estudio es el de probar a ambos detectores así como la etapa de corrección en una casuística más real alejada del caso ideal de los ECGs sintéticos. Se ha optado por un banco de pacientes transplantados porque son los que presentan menor variabilidad de ritmo cardíaco. De dicha confrontación esperamos medir hasta que punto los resultados teóricos se sostienen en un caso práctico.

En primer lugar, los resultados tanto del detector robusto como del detector sensible corroboran lo obtenido en el estudio con ECGs sintéticos ya que los cambios de morfología afectan en menor medida al primero que al segundo.

En segundo lugar, se ha procedido a introducir la fase de corrección para ambos detectores una vez ajustado el parámetro de la ventana. Por un lado, se ha observado que el detector poco robusto introduce para dichos pacientes una SDRR ligada a cambios de morfología del ECG cuya magnitud permite al corrector realizar su tarea eficientemente (no es ni demasiado próxima a cero ni demasiado grande). Por el contrario, la SDRR del detector robusto se debe más a la propia variabilidad de ritmo cardíaco del propio paciente que a cambios de morfología. Por el otro lado, de la comparación de los resultados corregidos de ambos detectores se ha visto que se ha logrado rebajar la magnitud de la SDRR del detector poco robusto hasta el punto de equiparla con el otro. Así pues, nuestra lectura es que la SDRR debida a cambios de morfología ha podido ser prácticamente cancelada por medio de la corrección por correlación.

Con todo, los resultados obtenidos en el caso práctico son coherentes con los resultados obtenidos en el caso teórico ideal. Tanto el comportamiento sin corrección del detector robusto y del detector sensible son consonantes en ambos casos así como la eficiencia de la corrección por correlación. El parámetro de la ventana óptima también se ha mostrado igual de válido en ambos contextos.

EN CONCLUSIÓN, los cambios de morfología afectan a los detectores en distinta medida. En el caso de los detectores robustos no será necesaria una etapa de corrección. Por el contrario, si se trata de un detector sensible la necesidad de corrección será mayor cuanto mayor sea la rotación debida a cambios de morfología. Asimismo, los cambios de morfología no afectarán del mismo modo a pacientes distintos.

Igualmente, subrayar que el error residual tras la corrección para detectores sensibles es de la misma magnitud que el error obtenido por detectores robustos dando a entender que los efectos de los cambios de morfología han sido minimizados. Además,

se trata de errores bastante pequeños y acotados (en nuestro caso menores a 2 ms). En consecuencia, en estudios ulteriores será tarea del investigador decidir si se pueden considerar despreciables o no en relación a la precisión con la que se desee trabajar.

FINALMENTE, añadir alguna sugerencia para futuras investigaciones que puedan contribuir a consolidar el presente. Por un lado, sería interesante profundizar en el estudio del cambio que sufren los índices de variabilidad del ritmo cardíaco cuando se cambia de detector de QRS. Además, podría investigarse la forma en que cambiaría la serie RR si en lugar de obtenerla de una única derivación se promediaran los resultados de n derivaciones. Sería conveniente también extrapolar nuestro estudio a otros casos como el de la respiración durante esfuerzos físicos o para los cambios de morfología del ECG atribuidos a cambios posturales del cuerpo. Por último, sería conveniente averiguar cuales son las causas de una dispar disminución de la relación señal a ruido entre dos pacientes distintos dado un mismo aumento de rotación de ejes. Una clasificación de los pacientes en tanto que sensibles o no a cambios de morfología del ECG podría facilitar estudios futuros.

14. Bibliografía

1. Fotiadis, Dimitris. Likas, Aristidis. Michalis, Lampros. Papaloukas, Costas. "Electrocardiogram (ECG): (automated diagnosis)" University of Ioannina. Ioannina, Greece. 2006
2. Redz, Anna. "Presentation and Analysis of Vector Electrocardiograms" 1998
3. Sörnmo, Leif. Laguna, Pablo. "Electrocardiogram (ECG) signal processing" Lund University Sweden, Zaragoza University Spain. 2006
4. <http://diec.unizar.es/~laguna/personal/publicaciones/caseib03-respiracion.pdf>
5. Rivera, María. "Excitación y conducción rítmica del corazón" Facultad de Ciencias y Filosofía UPCH
6. Akselrod, Solange. Pinhas, Itzik. R. Davrath, Linda. Shinar, Zvika. Toledo, Eran. "Heart Rate Variability (HRV)" Tel Aviv University, 2006
7. García, J. Aström, M. Sörnmo, L. Laguna, P. "ECG based detection of body position changes" Lund University, 2000
8. Li, Cui Wei. Zheng, Chong Xun. Tai, Chang Feng. "Detection of ECG Characteristic Points Using Wavelet Transforms" 1995
9. Poli, Riccardo. Cagnoni, Stefano. Valli, Guido. "Genetic Design of Optimjum Linear and Nonlinear QRS Detectors" 1995.
10. Köhler, Bert-Uwe. Hennig, Carsten. Orglmeister, Reinhold "The Principles of Software QRS Detection" 2002.
11. <http://www.physionet.org/physiobank/database/fantasia/>
12. <http://www.physionet.org/physiobank/database/ptbdb/>
13. Patrick S. Hamilton and Willis J. Tompkins. "*Quantitative Investigation of QRS Detection Rules Using the MIT/BIH Arrhythmia Database*". IEEE Transactions On Biomedical Engineering. Páginas 1157-1165. Vol. BME-33, Número 12. Diciembre de 1986.

15. Anexo

15.1 Detectores

Se ha incluido la versión de los detectores empleados en el apartado 3 y 4 de la memoria que son la esencia de las versiones posteriores. En los apartados siguientes sólo se ha adaptado la adquisición del ECG para su procesado (en un caso para los pacientes rotados y en el otro para los pacientes transplantados).

15.1.1 Detector Wavelet

Función “wl3”

```
function result=wl(fichero,frecMuestreo)
cd ('E:\ale\MATLAB7\work\ECG')
for j=1:10
    disp(j);
    if j<10
        fichero=['f1o0' num2str(j) '.dat'];
    else
        fichero=['f1o010.dat'];
    end
    fm=250;

    ecg=openfantasia1(fichero);

    [b,c]=butter(2,[20 60]/fm);
    ecg=filtfilt(b,c,ecg);

    dim=size(ecg);

    nmuestras=10000;%numero de muestras por bloque

    ps24=zeros(nmuestras,1);
    delta=[];
    numdelta=0;
    param=0.5;
    mother='haar';
    nbloques=round(dim(1)/nmuestras);

    for i=1:nbloques
        disp(i);
```

```

disp('de');
disp(nbloques);
if i<nbloques
    temp=ecg(1+(i-1)*nmuestras:i*nmuestras);
    [deltatemp numdeltastemp]=bloquewl(temp,fm,mother,param);
else
    temp=ecg(1+(i-1)*nmuestras:dim);
    [deltatemp numdeltastemp]=bloquewl(temp,fm,mother,param);
end
delta=[delta;deltatemp];
numdelta=numdeltastemp+numdelta;
end

ventana=100/1000*fm;
delta=refinar(delta,ecg,ventana);
rr=diferenciaentrelatidos(delta',numdelta);

[Se P]=performanceDetection(rr,numdelta);

DIR=['f2y'];
mkdir(DIR);
cd(DIR);
nombre=['variables' num2str(j)];
save(nombre,'rr','delta','numdelta','ecg','Se','P');
clear all
cd ..

end
end

```

Función “bloquewl”

```

function [delta numdelta]=bloquewl(ecg,fm,mother,param)

    interval=round(0.12*fm);

    scale21=2*fm/250;
    scale22=4*fm/250;
    scale23=8*fm/250;
    scale24=16*fm/250;

    %transformadas wl en cada escala
    WTat21=cwt(ecg,scale21,mother);
    WTat22=cwt(ecg,scale22,mother);
    WTat23=cwt(ecg,scale23,mother);
    WTat24=cwt(ecg,scale24,mother);

    WTat21=WTat21(:);
    WTat22=WTat22(:);

```

```

WTat23=WTat23(:);
WTat24=WTat24(:);

pico21=pico(WTat21,fm);
pico22=pico(WTat22,fm);
pico23=pico(WTat23,fm);
pico24=pico(WTat24,fm);

medpic21=mean(pico21);
medpic22=mean(pico22);
medpic23=mean(pico23);
medpic24=mean(pico24);

E1=param*medpic21;
E2=param*medpic22;
E3=param*medpic23;
E4=param*medpic24;

%paso 1: módulos máximos a escala 2^4,2^3,2^2,2^1...
umbral24=E4;
%puntos singulares a 2^4
ps24=find(abs(WTat24)>umbral24);
% ps24=optimizar(ps24);
ps23=zeros(length(ps24),1);
ps22=zeros(length(ps24),1);
ps21=zeros(length(ps24),1);

N=length(ps24);
disp('steps');
for i=1:N
    if mod(i,1000)==0
        disp(i);
    end
    ps23(i)=step(WTat23,ps24(i),E3,length(ecg),interval);
    if ps23(i)==0
        ps22(i)==0;
        ps21(i)==0;
    else
        ps22(i)=step(WTat22,ps23(i),E2,length(ecg),interval);
        if ps22(i)==0
            ps21(i)==0;
        else
            ps21(i)=step(WTat21,ps22(i),E1,length(ecg),interval);
        end
    end
end
end

index=find(ps21==0);
if ~isempty(index)
    ps21=quitarPuntos(ps21,index);

```

```

        ps22=quitarPuntos(ps22,index);
        ps23=quitarPuntos(ps23,index);
        ps24=quitarPuntos(ps24,index);
    end
    N=N-length(index);

    %elimination of Isolation and Redundant Modulus Maximum Lines
    %interval threshold = 120ms
    disp('isolation');
    [ps24,ps23,ps22,ps21,N]=isolation(WTat21,ps24,ps23,ps22,ps21,interval,N);
    disp('maxmin');

    [ps24,ps23,ps22,ps21,N]=maxMin(WTat23,ps24,ps23,ps22,ps21,N,'max',length(ecg),interval);

    [ps24,ps23,ps22,ps21,N]=maxMin(WTat23,ps24,ps23,ps22,ps21,N,'min',length(ecg),interval);

    %detector
    %   deltasTemp1=zeros(N,1);
    disp('detector picos')
    [delta numdelta]=detectorPicos(WTat21,ps21,E1,fm);

```

Función “pico”

```

function result = pico(input,fm)

dim=size(input);

total=round(dim(1)/fm)-1;
output=zeros(total,1);

for i=1:total
    output(i)=max(input(1+fm*(i-1):fm+fm*(i-1)));
end

result=output;

```

Función “step”

```

function [ps]=step(WT,psAnterior,umbral,nmuestras,interval)

    if psAnterior-interval+1<= 1
        a=1;
    else

```

```

        a=psAnterior-interval;
    end
    if psAnterior+interval > nmuestras
        b=nmuestras;
    else
        b=psAnterior+interval;
    end
    WTtreball=WT(a:b);

    maximos=find(abs(WTtreball)>umbral);
    if ~isempty(maximos)
        [maxmax,index]=max(abs(WTtreball(maximos)));
        temp=[];
        temp=maximos(1:index-1);
        temp=[temp;maximos(index+1:end)];

        indices=find(abs(WTtreball(temp))>maxmax/1.2);
        if length(indices)==0
            ps=maximos(index)+a-1;
        else
            [c,index]=min(abs(maximos(:)+a-psAnterior));
            ps=maximos(index)+a-1;
        end
    else
        ps=0;
    end
end
end

```

Función “quitarPuntos”

```

function [x]=quitarPuntos(vector,index)
x=zeros(length(vector)-length(index),1);
j=1;
a=1;
for i=1:length(vector)
    if a<=length(index)
        if i==index(a)
            a=a+1;
        else
            x(j)=vector(i);
            j=j+1;
        end
    else
        x(j)=vector(i);
        j=j+1;
    end
end
end
end

```

Función “isolation”

```
function
[ps24F,ps23F,ps22F,ps21F,N]=isolation(WTat21,ps24,ps23,ps22,ps21,interval,N)
index=[];
for i=1:N
    if mod(i,1000)==0
        disp(i);
    end

    if i==1
        if ps21(i+1)-ps21(i)>interval
            index=[index,i];
        end
    elseif i==N
        if ps21(i)-ps21(i-1)>interval
            index=[index,i];
        end
    else
        if ps21(i)-ps21(i-1)>interval & ps21(i+1)-ps21(i)>interval
            index=[index,i];
        end
    end
end
ps24=quitarPuntos(ps24,index);
ps23=quitarPuntos(ps23,index);
ps22=quitarPuntos(ps22,index);
ps21=quitarPuntos(ps21,index);
NF=N-length(index);
ps24F=ps24;
ps23F=ps23;
ps22F=ps22;
ps21F=ps21;
end
```

Función “maxMin”

```
function
[ps24F,ps23F,ps22F,ps21F,NF]=maxMin(WTat23,ps24,ps23,ps22,ps21,N,mm,nmuestras,interval)
descartes=[];

indexMax=find(WTat23(ps23)>0);
indexMin=find(WTat23(ps23)<0);
if mm=='max'
    indexMaxMin=zeros(length(indexMax),1);
    indexMaxMin(:)=indexMax(:);
else
```

```

    indexMaxMin=zeros(length(indexMin),1);
    indexMaxMin(:)=indexMin(:);
end

for i=1:length(indexMaxMin)
    if ps23(indexMaxMin(i))-interval+1 < 1
        a=1;
    else
        a=ps23(indexMaxMin(i))-interval+1;
    end
    if ps23(indexMaxMin(i))+interval > nmuestras
        b=nmuestras;
    else
        b=ps23(indexMaxMin(i))+interval;
    end
    index=[];
    index=[index,find(a<ps23&ps23<b&~(ps23==ps23(indexMaxMin(i))))];
    minimos=[];
    maximos=[];

    maximos=[maximos,find(WTat23(ps23(index))>0)];
    minimos=[minimos,find(WTat23(ps23(index))<0)];
    if mm=='max'
        maxmin=zeros(length(minimos),1);
    %    maxmin(:)=maximos(:);
        maxmin(:)=minimos(:);
    else
        maxmin=zeros(length(maximos),1);
    %    maxmin(:)=minimos(:);
        maxmin(:)=maximos(:);
    end

    distancias=zeros(length(maxmin),1);
    distancias(:)=maxmin(:)-ps23(indexMaxMin(i));

    %solo interesan los dos mínimos más cercanos al máximo
    if length(maxmin)>=2
        [temp,index]=sort(abs(distancias));
        A1=WTat23(maxmin(index(1)));
        A2=WTat23(maxmin(index(2)));
        L1=distancias(index(1));
        L2=distancias(index(2));

        if sign(L1)==sign(L2) %entonces están en el mismo lado...
            if abs(L1)>abs(L2)
                masCercano=index(2);
            else
                masCercano=index(1);
            end
            temp=[];
        end
    end
end

```

```

        temp=maxmin(1:masCercano-1);
        temp=[temp;maxmin(masCercano+1:end)];
        descartes=[descartes,temp'];
%        descartes=[descartes,quitarPuntos(maxmin,masCercano)'];
    else
        if abs(A1/L1) > abs(1.2*A2/L2)
            bueno=index(1);
            temp=maxmin(1:bueno-1);
            temp=[temp;maxmin(bueno+1:end)];
            descartes=[descartes,temp'];
%            descartes=[descartes,quitarPuntos(maxmin,bueno)'];
        else abs(A2/L2) > abs(1.2*A1/L1)
            bueno=index(2);
            temp=maxmin(1:bueno-1);
            temp=[temp;maxmin(bueno+1:end)];
            descartes=[descartes,temp'];
%            descartes=[descartes,quitarPuntos(maxmin,bueno)'];
        end
    end
end
end
if length(maxmin)==0
    descartes=[descartes,indexMaxMin(i)];
end
end
ps23=quitarPuntos(ps23,descartes);
ps22=quitarPuntos(ps22,descartes);
ps21=quitarPuntos(ps21,descartes);
ps24=quitarPuntos(ps24,descartes);
NF=N-length(descartes);
ps24F=ps24;
ps23F=ps23;
ps22F=ps22;
ps21F=ps21;
end

```

Función “detectorPicos”

```

function [x ndeltas]=detectorPicos(WT,ps,A,fm)
    x=zeros(length(WT),1);
    temp=[];
    x1=zeros(length(WT),1);
    umbral=0.3*A;
    ndeltas=0;

    for i=1:length(ps)
        if i==1
            if abs(WT(ps(i)))>umbral

```

```

        x1(ps(i))=1;
        ndeltas=ndeltas+1;
    end
end
if i>1
    if abs(WT(ps(i)))>umbral && ps(i)>ps(i-1)+fm/3
        x1(ps(i))=1;
        ndeltas=ndeltas+1;
    end
end
%actualizar umbral
if abs(WT(ps(i)))>2*A
    A=A;
else
    A=0.875*A+0.125*abs(WT(ps(i)));
end
umbral=0.3*A;

end
%asegurar picos ancho 1
x=adeltazar(x1);
end

```

Función “refinar”

```

function [trenSalida]=refinar(tren,ecg,intervalo)
long=length(tren);
trenSalida=zeros(1,long);
punteros=find(tren==1);
for i=1:length(punteros)
    index=punteros(i);
    if ~(index+intervalo<length(ecg))
        [temp,I]=max(ecg(index-intervalo:~end));
        trenSalida(I+index-intervalo-1)=1;
    elseif ~(index-intervalo>0)
        [temp,I]=max(ecg(1:index+intervalo));
        trenSalida(I+index-intervalo-1)=1;
    else
        [temp,I]=max(ecg(index-intervalo:index+intervalo));
        trenSalida(I+index-intervalo-1)=1;
    end
end
end

```

Función “diferenciaentrelatidos”

```

function result = diferenciaentrelatidos(input,numdeltas)

```

```

dimin=size(input);
j=1;
a=0;
b=0;
an=0;
bn=0;
k=zeros(numdeltas,1);

for i=1:dimin(1)
    if input(i)==1 && a==0
        a=1;
        an=i;
    elseif input(i)==1 && a==1
        b=1;
        bn=i;
    end
    if a==1 && b==1
        k(j)=bn-an;
        j=j+1;
        an=bn;
        bn=0;
        a=b;
        b=0;
    end
end

result=k;
disp('Diferencia entre QRS detectadas');
end

```

Función “performanceDetection”

```
function [Se P] = performanceDetection(rr,numdeltas)
```

```

    FP=0;
    FN=0;
    media=mean(rr);
    for i=1:length(rr)
        if rr(i)>1.3*media
            FN=FN+1;
        end
        if rr(i)<0.7*media
            FP=FP+1;
        end
    end
    TP=numdeltas-FP-FN;
    Se=TP/(TP+FN);
    P=TP/(TP+FP);

```

end

15.1.2 Detector Genetic Algorithm

Función “GA”

```
function result=GA(fichero,fm)

    cd ('E:\ale\MATLAB7\work\ECG')

    fichero=['f1y01.dat'];

    fm=250;

    ecg=openfantasia1(fichero);

    nmuestras=length(ecg);
    tipofiltrado='A';
    y=zeros(nmuestras,1);

    switch tipofiltrado
        case 'A'
            disp('filtrado');
            y=polynomialA(ecg,nmuestras);
            disp('detección');
            [Ystart Ymax Ymin]=findYs(y,fm);
            [tren numdeltas]=maximaDetector(y,fm,tipofiltrado,Ystart,Ymax,Ymin);
        case 'B'
            disp('filtrado');
            y=polynomialB(ecg,nmuestras,fm);
            disp('detección');
            [maximos numdeltas]=maximaDetector(y,fm,tipofiltrado);
        otherwise
            disp('unknown method');
    end
    rr=diferenciaentrelatidos(tren,numdeltas);

    figure(1);
    plot(rr);
    figure(2);
    plot([maximos ecg(1:nmuestras)/(2.5*10^4)]);
    save('variables','rr','maximos','numdeltas','ecg');
```

Función “polynomialA”

```
function y = polynomialA(x,nmuestras)
%Quasi-Linear Filters with Consecutive Samples
%M=1,N=10

%parameter definition
a1=-7.13024;
a2=-7.94744;
a3=3.04609;
a4=4.77475;
a5=9.22591;
a6=7.31803;
a7=9.37990;
a8=-8.26923;
a9=-9.57643;
a10=-0.62212;
y=zeros(nmuestras,1);

for i=1:1:nmuestras
    y(i)=a1*x(i-1)+a2*x(i-2)+a3*x(i-3)+a4*x(i-4)+...
        a5*x(i-5)+a6*x(i-6)+a7*x(i-7)+a8*x(i-8)+a9*x(i-9)+a10*x(i-10);
end
```

Función “findYs”

```
function [Ystart Ymax Ymin] = findYs(y,fm)
disp('en findYs');
vector=zeros(round(length(y)/1000),1);
N=length(vector);
for i=1:N
    if i<N
        a=1+(i-1)*1000;
        b=i*1000;
        vector(i)=max(y(a:b));
    else
        a=(i-1)*1000;
        vector(i)=max(y(a:length(y)));
    end
end
Ystart=0.7*mean(vector);
Ymax=0.9*mean(vector);
Ymin=0.4*mean(vector);
end
```

Función “maximaDetector”

```
function [y numdeltas]=maximaDetector(vector,fm, caso, Ystart, Ymax, Ymin)
```

```
%parameters
switch caso
    case 'A'
        C=round(25/120*fm);
        alfa=0.00216;
        beta=0.50137;
        gamma=0.93127;
    case 'B'
        C=round(31/120*fm);
        alfa=0.00336;
        beta=0.39331;
        gamma=0.87430;
    case 'C'
        C=round(38/120*fm);
        alfa=0.00384;
        beta=0.23277;
        gamma=0.69204;
end

N=length(vector);
y=zeros(N,1);
umbral=Ystart;
lastQRS=0;
numdeltas=0;

for i=1:N
    if vector(i)>umbral
        if lastQRS==0
            lastQRS=i;
            y(i)=1;
            numdeltas=numdeltas+1;
        else
            if i-lastQRS>C
                lastQRS=i;
                y(i)=1;
                numdeltas=numdeltas+1;
            end
        end
    end
    %actualizamos umbral
    if i>1
        temp=(1-alfa)*umbral-beta*(umbral-gamma*vector(i-1));
        umbral=max(Ymin,min(Ymax,temp));
    end
end
```

Función “diferenciaentrelatidos” → ver apartado 15.1.1

Función “performanceDetection” → ver apartado 15.1.1

15.1.3 Detector Hilbert

Función “detH”

```
function result = detectordehilbert (fichero)
%Detector de QRS basado en la transformada de Hilbert

cd ('E:\ale\MATLAB7\work\ECG')

fichero=['f1o01.dat'];

fm=250;

ecg=openfantasia1(fichero);

dim=size(ecg);
ecgtf=ecg(1:dim(1));

%Preprocesado de señal(Filtrado paso banda)
[b,c]=butter(2,[20 60]/250);
ecgtf=filtfilt(b,c,ecgtf);
disp('Señal Filtrada');

%Procesado de la señal. Transformada de Hilbert
hecg=hilbert(ecgtf-mean(ecgtf));
abhecg=abs(hecg);
disp('Señal Procesada con la Transformada de Hilbert');

%deteccion con umbral
dt = estimacionumbral (abhecg,0.5,0.2,fm);
[tren numdeltas]=sobrepasaumbral3 (abhecg,dt,fm);

%Para evitar los retrasos producidos por los filtros
%Se retrasan 25 muestras con Butterworth y 39 con los 2 filtrados
%delta=retrasadeltas(delta,25);

%muestras entre latidos
rr=diferenciaentrelatidos(tren,numdeltas);

[Se P]=performanceDetection(rr,numdeltas);
```



```

    DIR=['f2y'];
    mkdir(DIR);
    cd(DIR);
    nombre=['variables' num2str(i)];
    save(nombre,'rr','tren','numdeltas','ecg','Se','P','abhecg');
    clear all
    cd ..
end

```

Función “pico”

```

function result = pico(input, fm)

dim=size(input);

total=round(dim(1)/fm)-1;
output=zeros(total,1);

for i=1:total
    output(i)=max(input(1+fm*(i-1):fm+fm*(i-1)));
end

result=output;

```

Función “estimacionumbral”

```

function result = estimacionumbral (input,a,b,fm)

%código de la derivada respecto el tiempo

dim=size(input);
output=zeros(dim(1),1);
picos=pico(input,fm); %Deteccion de nivel de señal en los picos
dim2=size(picos);
ple=zeros(dim2(1)+1,1); %peak level estimation
v=round(dim2(1)/10);
ple(1)=mean(picos(1:v));
dt=zeros(dim2(1)+1,1); %umbral de deteccion
dt(1)=0;

for i=2:dim2(1)+1
    ple(i)=(1-a)*ple(i-1)+a*picos(i-1); %Algoritmo adaptativo de nivel de señal en el
    pico
end

npl=median(input); %noise peak level

```

```

for i=1:dim2(1)+1
    dt(i)=npl+b*(ple(i)-npl);
end
result = dt;

disp('Umbrales Estimados');

end

```

Función “sobrepasaumbral3”

```

function [delta,numerodedeltas] = sobrepasaumbral2 (input,dt,fm)

dimin=size(input);
dimdt=size(dt);
interv=round(dimin(1)/fm)-1; %Division de todo el QRS por latidos
a=find(input(1:fm)>dt(1)); %Se guardaran las muestras que sobrepasan el umbral en
cada latido
s=a; %Se guardaran en s todas las muestras del QRS que sobrepasan el umbral
adaptativo

for i=2:interv
    a=find(input(1+fm*(i-1):fm*i)>dt(i)); %Buscar que valores de cada latido superan el
umbral
    a=a+1+fm*(i-1); %Numero de indice segun el numero de latido
    s=[s;a]; %Volcado de datos del vector que tiene solo un latido al que contiene todos
end

i=1;
j=1;
k=1;
dims=size(s);

while i<dims(1)

    b=s(i); %Indice donde superamos por primera vez el umbral
    while (s(i+1)==s(i)+1) && (i<dims(1)-1)
        i=i+1;
    end
    c=s(i); %Indice donde bajamos por debajo del valor del umbral
    m(j)=(b+c)/2; %Media de los 2 valores anteriores
    m(j)=round(m(j)); %Redondeo para evitar una posicion de indice decimal
    j=j+1;
    i=i+1;
end

j=1;

```

```

deltas=zeros(dimin(1),1);

dimm=size(m);

%Asignacion de cada posicion anterior a una delta
i=1;
for i=1:dimin(1)
    if j<=dimm(2)
        if i==m(j)
            deltas(i)=1;
            j=j+1;
        end
    end
end

i=1;
numerodedeltas=0;

%Para obtener solo una delta por QRS y de este modo discriminar las demás
while i<=dimin(1)
    k=1;
    if deltas(i)==1
        i1=i;
        i2=1;
        deltas(i)=0;
        while k<=round(80*fm/250) && i<dimin(1) %Definición del intervalo durante el
            cual solo puede haber una delta
                i=i+1;
                k=k+1;
                if deltas(i)==1
                    i1=i1+i;
                    i2=i2+1;
                    deltas(i)=0;
                end
            end
            i1=i1/i2;
            i1=round(i1);
            deltas(i1)=1;
            numerodedeltas=numerodedeltas+1;
        end
    end
    i=i+1;
end

delta=deltas;

disp('Picos de QRS detectados');

end

```

Función “retrasadeltas”

```
function [x] =retrasaDeltas(ecg,tren,T)
x=zeros(length(ecg),1);
intervalo=round(T/3);
a=find(tren==1);
i=1;
while a(i)<intervalo+1
    i=i+1;
end
[C,index]=max(ecg(a(i)-intervalo:a(i)+intervalo));
index=index+a(i)-intervalo-1;
tren=tren(a(i):end);
x(index:length(tren)+index-1)=tren(1:end);
end
```

Función “diferenciaentrelatidos” → ver apartado 15.1.1

Función “perfomanceDetection” → ver apartado 15.1.1

15.1.4 Detector Zero Crossings

Función “zero”

```
function result=zero(input,fm)
cd ('E:\ale\MATLAB7\work\ECG')

    fichero=['f1o01.dat'];

    fm=250;

    ecg=openfantasia1(fichero);

    fm=250;

    ventana=round(50/250*fm);

    %adquisicion ECG
    ecg=openfantasia1(fichero);
    ecg=ecg-mean(ecg);

    nmuestras=100000;
    %filtrado
    disp('filtrado butt');
```

```

[b,c]=butter(2,[20 60]/fm);
ecgF=filtfilt(b,c,ecg);
T=buscarPeriodo(ecgF,fm) ;
%añadimos secuencia de alta frecuencia
%calcula amplitud b(n)
disp('calcula b(n)');
b=zeros(nmuestras,1);
b=modulusAveragebis(ecgF,nmuestras,300);

y=zeros(nmuestras,1);
y=ecgF(1:nmuestras)+b;

%construccion de z(n) con promediados de zeros en una ventana
disp('promediado de cruces por cero');
z=zerocrossingsbis(y,ventana);

%convertir mínimos de z en máximos
z=abs(z-max(z));

%umbral adaptativo
disp('umbral adaptativo');
umbral=[];
umbral=estimacionumbral(z,0.5,0.2,fm);

%detector R
disp('buscamos Rs');
tren=[];
[tren,numdeltas]=sobrepasaumbral3(z,umbral,fm);
tren=retrasadeltas(ecg,tren,T);
rr=diferenciaentrelatidos(tren,numdeltas);

[Se P]=performanceDetection(rr,numdeltas);

DIR=['f2yVENT=60'];
mkdir(DIR);
cd(DIR);
nombre=['variables' num2str(i)];
save(nombre,'rr','y','tren','numdeltas','ecg','Se','P','b','y','z');
clear all
cd ..

end

```

Función “buscarPeriodo”

```

function T=buscarPeriodo(ecg,fm)

picos=[];
[b,c]=butter(2, [20 60]/1000);

```

```

ecgFilt=filtfilt(b,c,ecg(round(2000*1000/fm):round(6000*1000/fm)));

mayorQumbral=find(ecgFilt>0.6*max(ecgFilt));

for i=2:length(mayorQumbral)-1
    if ecgFilt(mayorQumbral(i-1))<ecgFilt(mayorQumbral(i))&...
        ecgFilt(mayorQumbral(i))>ecgFilt(mayorQumbral(i+1))
        picos=[picos mayorQumbral(i)];
    end
end

i=2;
while abs(picos(1)-picos(i))<round(200*1000/fm)
    i=i+1;
end

T=abs(picos(1)-picos(i));

end

```

Función “modulusAveragebis”

```

function [m]=modulusAverage(vector,N,ventana)
m=zeros(N,1);
for i=1:N
    if mod(i,100000)==0
        disp(i);
    end
    if i<N-ventana+1
        if mod(i,2)==0
            m(i)=mean(abs(vector(i:i+ventana)));
        else
            m(i)=-mean(abs(vector(i:i+ventana)));
        end
    else
        if mod(i,2)==0
            m(i)=mean(abs(vector(i:N)));
        else
            m(i)=-mean(abs(vector(i:N)));
        end
    end
end
end
end

```

Función “zerocrossingsbis”

```

function [x]=zerocrossings(vector,interval)

```

```

N=length(vector);
x=zeros(N,1);
for i=1:N
    count=0;
    if i<interval
        for j=1:i+interval-1
            if ~(sign(vector(j))==sign(vector(j+1)))
                count=count+1;
            end
        end
    elseif i>N-interval
        for j=i-interval+1:N-1
            if ~(sign(vector(j))==sign(vector(j+1)))
                count=count+1;
            end
        end
    else
        for j=i-interval+1:i+interval-1
            if ~(sign(vector(j))==sign(vector(j+1)))
                count=count+1;
            end
        end
    end
    x(i)=count;
end
end

```

Función “estimacionumbral” → ver apartado 15.1.3

Función “sobrepasaumbral3” → ver apartado 15.1.3

Función “retrasadeltas” → ver apartado 15.1.3

Función “diferenciaentrelatidos” → ver apartado 15.1.1

Función “perfomanceDetection” → ver apartado 15.1.1

15.2 Creación de sintéticos

Se ha incluido la primera versión en el anexo. Las diferentes variaciones se basan en dicho patrón sólo que se varía el número de pacientes a rotar o la magnitud a la que se les desea rotar.

Función “main”

```
function m=main()
fm=1000;
N=500;
fichero=[];
for ii=6:10
    cd('E:\ale\MATLAB7\work\fase2\registros');
    paciente=ii;
    DIR0='E:\ale\MATLAB7\work\fase2\registros\';
    DIR0=[DIR0 num2str(paciente)];
    cd(DIR0);
    mat=ls;
    dim=size(mat);
    numficheros=dim(1)-2;
    for iii=1:numficheros

        fichero=mat(iii+2,1:dim(2));

        [x,y,z]=abrir(paciente,fichero);

        T=buscarPeriodo(x,fm);
        xx=crearSintetico(x,N,fm,T);
        yy=crearSintetico(y,N,fm,T);
        zz=crearSintetico(z,N,fm,T);
        for i=1:2
            for j=1:7
                switch j
                    case 1
                        fas1=5*i*1*sin(2*pi*0.2/fm*(1:length(xx)));
                        fas2=5*i*0*sin(2*pi*0.2/fm*(1:length(yy)));
                        fas3=5*i*0*sin(2*pi*0.2/fm*(1:length(zz)));
                    case 2
                        fas1=5*i*0*sin(2*pi*0.2/fm*(1:length(xx)));
                        fas2=5*i*1*sin(2*pi*0.2/fm*(1:length(yy)));
                        fas3=5*i*0*sin(2*pi*0.2/fm*(1:length(zz)));
                    case 3
                        fas1=5*i*1*sin(2*pi*0.2/fm*(1:length(xx)));
                        fas2=5*i*1*sin(2*pi*0.2/fm*(1:length(yy)));
                        fas3=5*i*0*sin(2*pi*0.2/fm*(1:length(zz)));
```

```

    case 4
        fas1=5*i*0*sin(2*pi*0.2/fm*(1:length(xx)));
        fas2=5*i*0*sin(2*pi*0.2/fm*(1:length(yy)));
        fas3=5*i*1*sin(2*pi*0.2/fm*(1:length(zz)));
    case 5
        fas1=5*i*1*sin(2*pi*0.2/fm*(1:length(xx)));
        fas2=5*i*0*sin(2*pi*0.2/fm*(1:length(yy)));
        fas3=5*i*1*sin(2*pi*0.2/fm*(1:length(zz)));
    case 6
        fas1=5*i*0*sin(2*pi*0.2/fm*(1:length(xx)));
        fas2=5*i*1*sin(2*pi*0.2/fm*(1:length(yy)));
        fas3=5*i*1*sin(2*pi*0.2/fm*(1:length(zz)));
    case 7
        fas1=5*i*1*sin(2*pi*0.2/fm*(1:length(xx)));
        fas2=5*i*1*sin(2*pi*0.2/fm*(1:length(yy)));
        fas3=5*i*1*sin(2*pi*0.2/fm*(1:length(zz)));
    otherwise
        fas1=0*i*sin(2*pi*0.2/fm*(1:length(xx)));
        fas2=0*i*sin(2*pi*0.2/fm*(1:length(yy)));
        fas3=0*i*sin(2*pi*0.2/fm*(1:length(zz)));
end

[xr,yr,zr]=rotecg(xx,yy,zz,fas1,fas2,fas3);

if j==1&i==1
    cd('E:\ale\MATLAB7\work\fase2\muy rotados')
    DIR=['E:\ale\MATLAB7\work\fase2\muy rotados\paciente' ...
        num2str(paciente)];
    mkdir(DIR);
end
if j==1&i==1
    DIR=[DIR '\ fichero'];
    mkdir(DIR);
end

cd(DIR);

if j==1
    nombre=[num2str(paciente) '_' num2str(0)];
    save(nombre,'xx','yy','zz');
end
nombre=[num2str(paciente) '_' num2str(i) '_' num2str(j)];
save(nombre,'xr','yr','zr');
cd('E:\ale\MATLAB7\work\fase2')
end
end
end
end
end

```

Función “abrir”

```
function [x,y,z]=abrir(paciente,fichero)

DIR=['E:\ale\MATLAB7\work\fase2\registros\'];
DIR=[DIR num2str(paciente)];
cd(DIR)
fid=fopen(fichero);
a=fread(fid,'short');
fclose(fid);
cd('E:\ale\MATLAB7\work\fase2')

x=zeros(round(length(a)/3),1);
y=zeros(round(length(a)/3),1);
z=zeros(round(length(a)/3),1);

x(1:end)=a(1:3:end);
y(1:end)=a(2:3:end);
z(1:end)=a(3:3:end);

end
```

Función “buscarPeriodo” → apartado 15.1.4

Función “crearSintético”

```
function y = crearSintetico(ecg,N,fm,T)

sintetico=[];
entorno=round(50*1000/fm);

puntoInicial=buscarZonaPlana(ecg,entorno,fm);

pulsoMadre=zeros(T,1);
pulsoMadre(1:end)=ecg(puntoInicial:puntoInicial+T-1);
a=abs(ecg(puntoInicial)-ecg(puntoInicial+T));
signo=sign(ecg(puntoInicial)-ecg(puntoInicial+T));
m=a/T;
recta=zeros(T,1);
for i=1:T
    recta(i)=ecg(puntoInicial+T)+signo*m*i;
end

pulsoMadre(1:end)=pulsoMadre(1:end)+recta(1:end);

for i=1:N
    sintetico=[sintetico pulsoMadre(1:end)];
end
```

```

end

y=sintetico;
end

```

Función “buscarZonaPlana”

```

function y = buscarZonaPlana(ecg,entorno,fm)

count=zeros(5*fm,1);
p=rand(1)*10000+rand(1)*1000;
punto=round(p*1000/fm);
n=5*fm;
for i=1:n
    media=mean(ecg(punto+i-entorno:punto+i+entorno));
    for j=1:2*entorno+1
        count(i)=count(i)+abs(ecg(punto+i+j)-media);
    end
end

[C,I]=min(count);
y=I+punto;

end

```

15.3 Eliminación de artefactos

Función “precorrecciónRRspararesults”

```

function main=precorreccionRR()
fm=1000;
ventana=round(100*fm/1000);
tabla=zeros(294,1);

for ili=1:9
    switch ili
        case 1
            detector='energy';
        case 2
            detector='hilbert';
        case 3
            detector='zero crossing';
        case 4
            detector='genetic algorithm';
        case 5

```

```

        detector='trahanias';
    case 6
        detector='ruha';
    case 7
        detector='length';
    case 8
        detector='wavelet';
    otherwise
        detector='hamilton';
end
for a=1:10
    s=['paciente' int2str(a)];
    disp(s);
    paciente='paciente';
    paciente=[paciente num2str(a)];

    DIR=['E:\ale\MATLAB7\work\fase2\results\' detector '\'];
    DIR=[DIR paciente];
    cd(DIR);
    mat=ls;
    dim=size(mat);
    numficheros=dim(1)-2;
    for b=1:numficheros
        DIR1=[DIR '\' mat(b+2,1:dim(2))];

        for c=1:15
            cd(DIR1);
            if c==1
                fichero=[int2str(a) '_0.mat'];
            else
                if c<9
                    temp=c-1;
                    fichero=[int2str(a) '_' '1' '_' int2str(temp)...
                        '.mat'];
                else
                    temp=c-8;
                    fichero=[int2str(a) '_' '2' '_' int2str(temp)...
                        '.mat'];
                end
            end
        end
        cd(fichero);
        fm=1000;
        ventana=round(50/250*fm);
        if c==1
            temp1='xx.mat';
            temp2='yy.mat';
            temp3='zz.mat';
        else
            temp1='xr.mat';
            temp2='yr.mat';
        end
    end
end

```

```

    temp3='zr.mat';
end
open(temp1);
rrx=ans.rr;
ecgx=ans.ecg;
trenx=ans.tren;
numdeltasx=ans.numdeltas;
Tx=round(length(ecgx)/500);
open(temp2);
rry=ans.rr;
ecgy=ans.ecg;
treny=ans.tren;
numdeltasx=ans.numdeltas;
Ty=round(length(ecgy)/500);
open(temp1);
rrz=ans.rr;
ecgz=ans.ecg;
trenz=ans.tren;
numdeltasx=ans.numdeltas;
Tz=round(length(ecgz)/500);
[rrx,numdeltasx]=eliminarFP(rrx,Tx);
[rry,numdeltasx]=eliminarFP(rry,Ty);
[rrz,numdeltasx]=eliminarFP(rrz,Tz);
varTemp=['E:\ale\MATLAB7\work\fase2\results 5\'...
    detector ' sin artefactos'];
cd(varTemp);

temp11=['paciente' int2str(a)];
if b==1 & c==1
    mkdir(temp11);
end
temptemp=['.\' temp11];
cd(temptemp);

temp22=[mat(b+2,1:dim(2))];
if c==1
    mkdir(temp22);
end
temptemp=['.\' temp22];
cd(temptemp);

mkdir(fichero);
temptemp=['.\' fichero];
cd(temptemp);
    ecg=ecgx;
    tren=trenx;
    numdeltas=numdeltasx;
    rr=rrx;
    save('xr','ecg','tren','numdeltas','rr');
```

```

        ecg=ecgy;
        tren=treny;
        numdeltas=numdeltasy;
        rr=rry;
        save('yr','ecg','tren','numdeltas','rr');

        ecg=ecgz;
        tren=trenz;
        numdeltas=numdeltasz;
        rr=rrz;
        save('zr','ecg','tren','numdeltas','rr');

    end

end

end

end

```

Función “eliminarFP”

```

function [rrout,numdeltas]=eliminarFP(rr,T)
long=length(rr);
rrout=[];
rrout1=[];
cont=0;
for i=1:long
    if rr(i)<0.8*T
        cont=cont+rr(i);
        if cont>0.8*T
            rrout1=[rrout1,cont];
            cont=0;
        end
    else
        if cont>0.8*T
            rrout1=[rrout1,cont];
            rrout1=[rrout1,rr(i)];
            cont=0;
        else
            rrout1=[rrout1,rr(i)];
        end
    end
end
temp=find(rrout1<1.2*T);
rrout=rrout1(temp);
numdeltas=length(rrout);

```

15.4 Corrección por correlación

Se ha incluido la versión básica del corrector. Las modificaciones posteriores sólo han consistido en variaciones en función de la adquisición de los diferentes bancos de ECGs o variaciones del tamaño de la ventana.

Función “corrector”

```
function main=corrector()
fm=1000;
ventana=round(100*fm/1000);

for i=3:3
    switch i
        case 1
            detector='energy';
        case 2
            detector='hilbert';
        case 3
            detector='zero crossing revisado';
        case 4
            detector='genetic algorithm';
        case 5
            detector='length';
        case 6
            detector='ruha';
        case 7
            detector='trahanas';
        case 8
            detector='wavelet';
        case 9
            detector='correlation';
        otherwise
            detector='hamilton';
    end
    for a=1:10
        s=['paciente' int2str(a)];
        disp(s);
        paciente='paciente';
        paciente=[paciente num2str(a)];

        DIR=['E:\ale\MATLAB7\work\fase2\results 2\' detector ''];
        DIR=[DIR paciente];
        cd(DIR);
        mat=ls;
        dim=size(mat);
```

```
numficheros=dim(1)-2;

for b=1:1
    DIR1=[DIR '\ mat(b+2,1:dim(2))];

    for c=1:6
        cd(DIR1);
        fichero=['paciente' int2str(a) 'girado' int2str(c) '.mat'];
        cd(fichero);
        try
            temp='xr.mat';
            open(temp);
        catch
            temp='xx.mat';
            open(temp);
        end
        rr=ans.rr;
        ecg=ans.ecg;
        [bBb,cCc]=butter(2,[10 60]/fm);
        ecg=filtfilt(bBb,cCc,ecg);
        tren=ans.tren;
        l=length(rr);
        rc=[];
        n=[];
        %busco primer pico
        bbb=find(tren(1:round(4000*fm/1000))==1);
        ii=1;
        while bbb(ii)<ventana+1
            ii=ii+1;
        end
        [nnn,index]=max(ecg(bbb(ii)-ventana:bbb(ii)+ventana));
        index=index+bbb(ii)-ventana-1;
        r=cumsum(rr)+index;
        r=[index;r];
        ii=1;
        while r(ii)<ventana+1
            ii=ii+1;
        end
        mask=ecg(r(ii)-ventana:r(ii)+ventana);

        mask=detrend(mask);
        for i=1:length(r)
            if r(i)+ventana <length(ecg) & r(i)-ventana>0
                ecgt=detrend(ecg(r(i)-ventana:r(i)+ventana));
                xc=xcorr(mask,ecgt);
                [m,n(i)]=max(xc);
            end
        end

        r=r(1:length(n));
```

```
rc=r-n'+2*ventana+1;

cd('E:\ale\MATLAB7\work\fase2\corregs v=100 f p10 _y\');

temp1=[detector '\paciente' int2str(a)];
if b==1 & c==1
    MKDIR(temp1);
end
temptemp=['.\' temp1];
cd(temptemp);

temp2=['fichero' mat(b+2,1:dim(2))];
if c==1
    MKDIR(temp2);
end
temptemp=['.\' temp2];
cd(temptemp);
rrCorregido=diff(rc);
sigmaCorregido=std(rrCorregido(5:end-20));
sigma=std(rr(5:end-20));
temperatura=['corregidos' int2str(c)];
save(temperatura,'rrCorregido','rr','index','sigma',...
    'sigmaCorregido');
end
end
end
end
```